

*Fig. 1*  
*Prior Art*

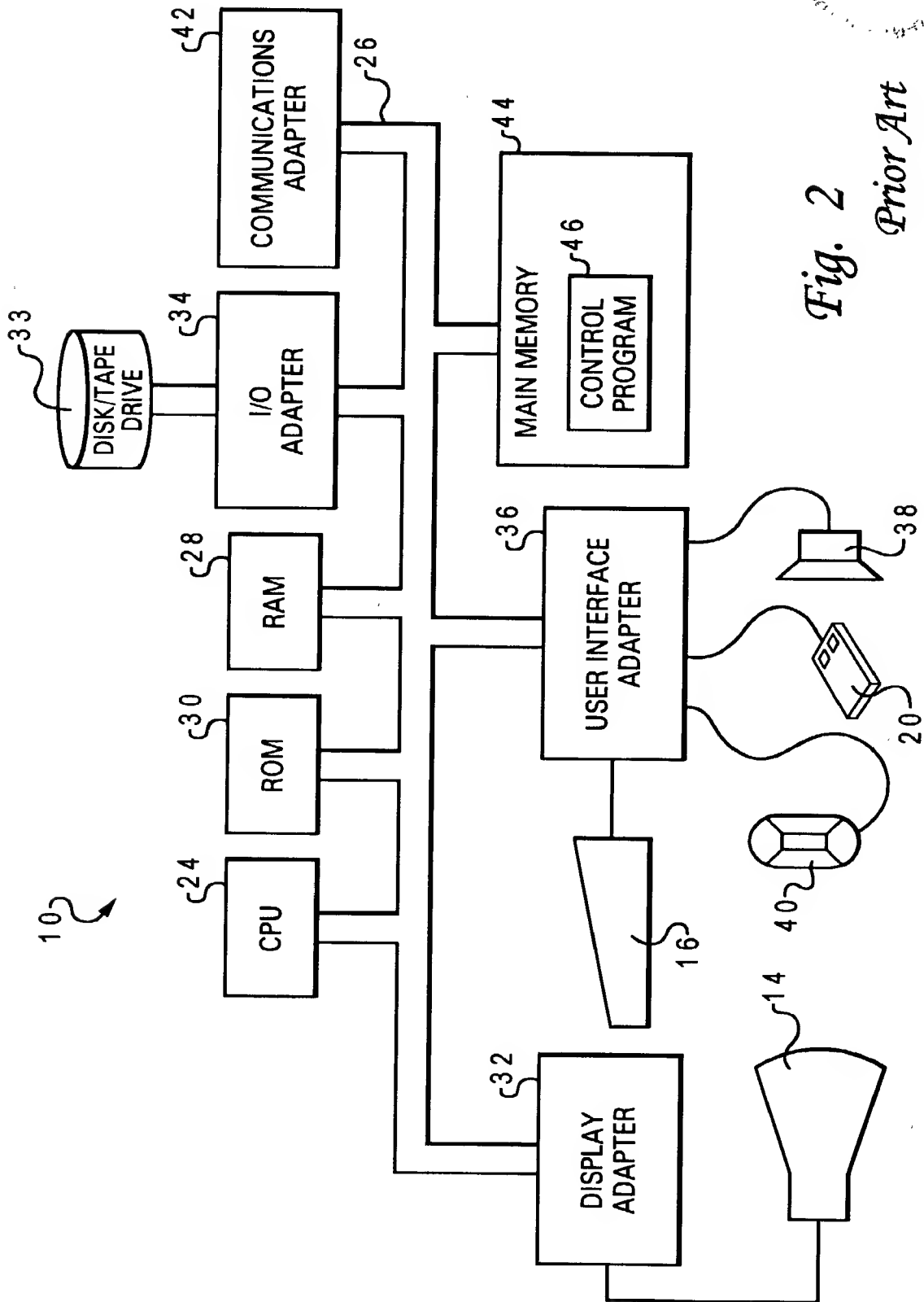


Fig. 2  
Prior Art

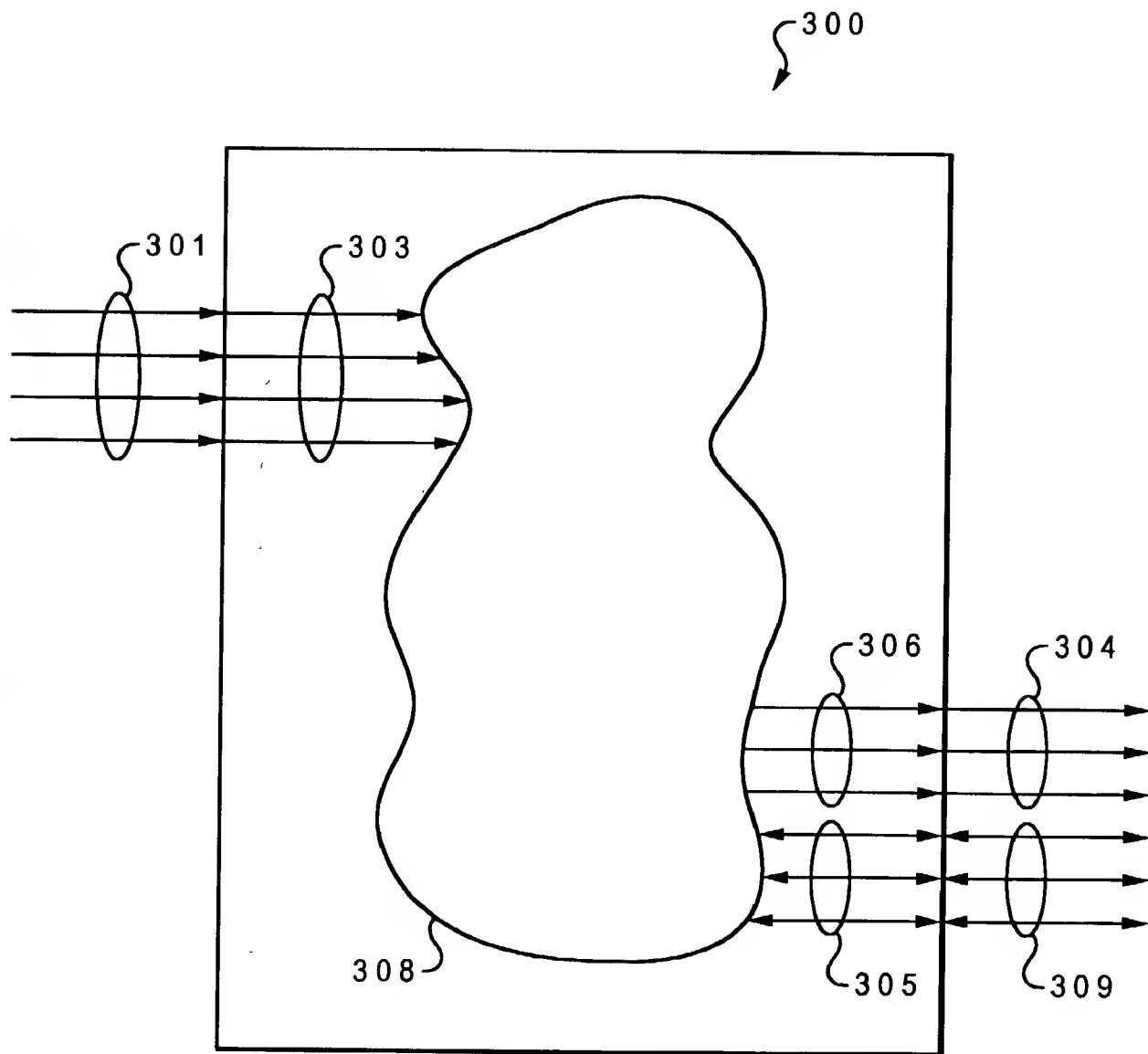
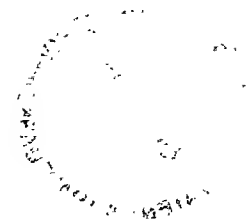
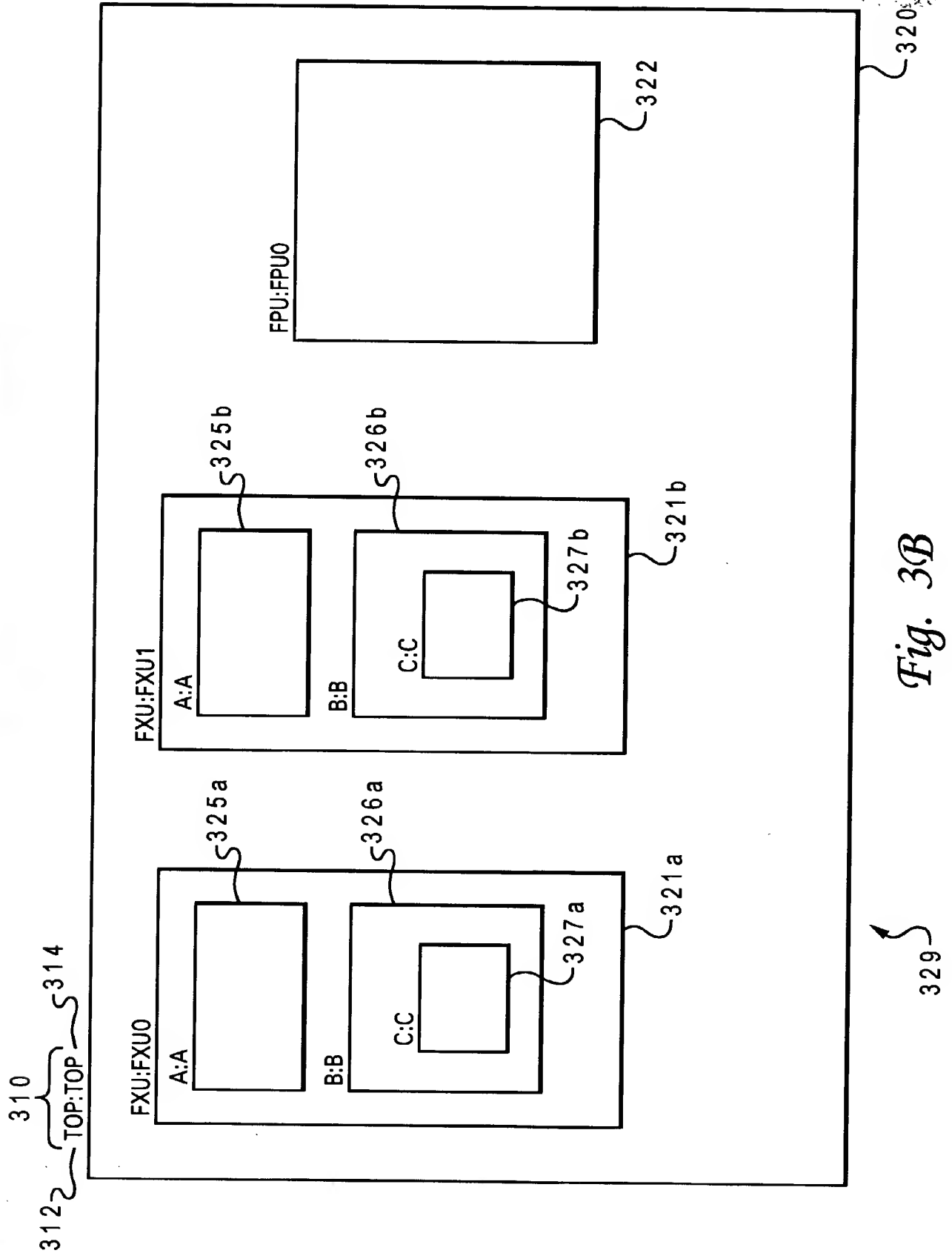
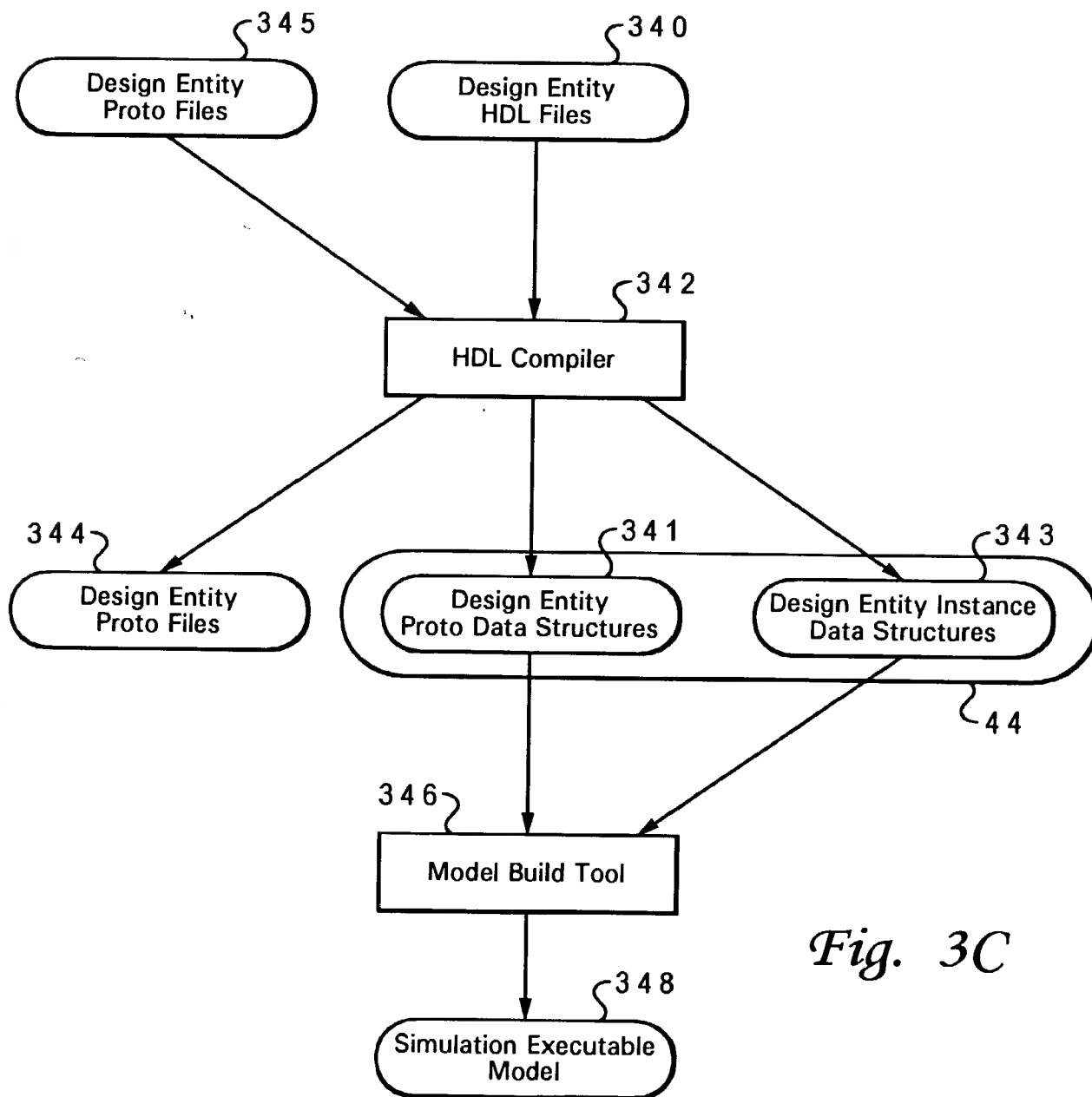


Fig. 3A





*Fig. 3C*

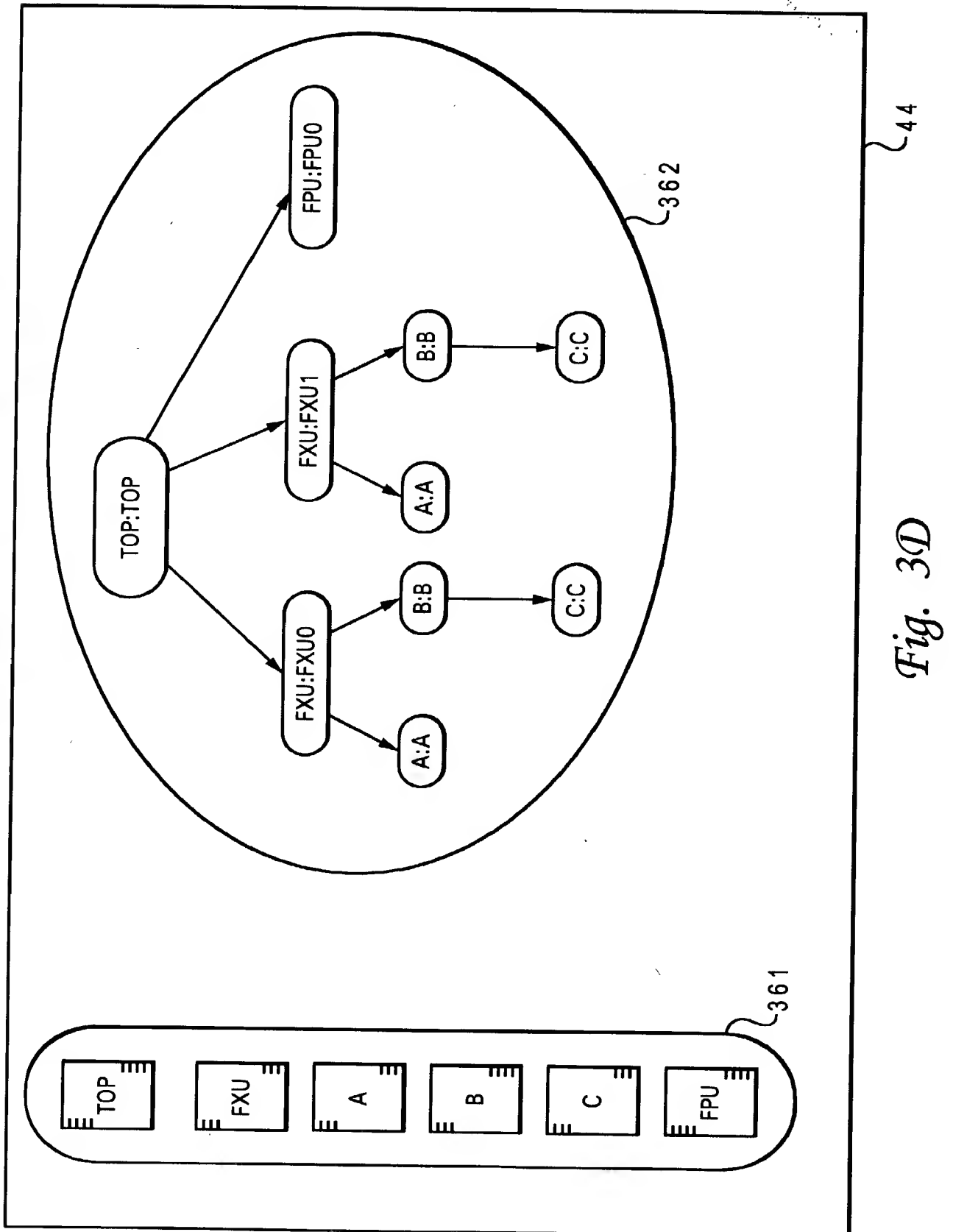
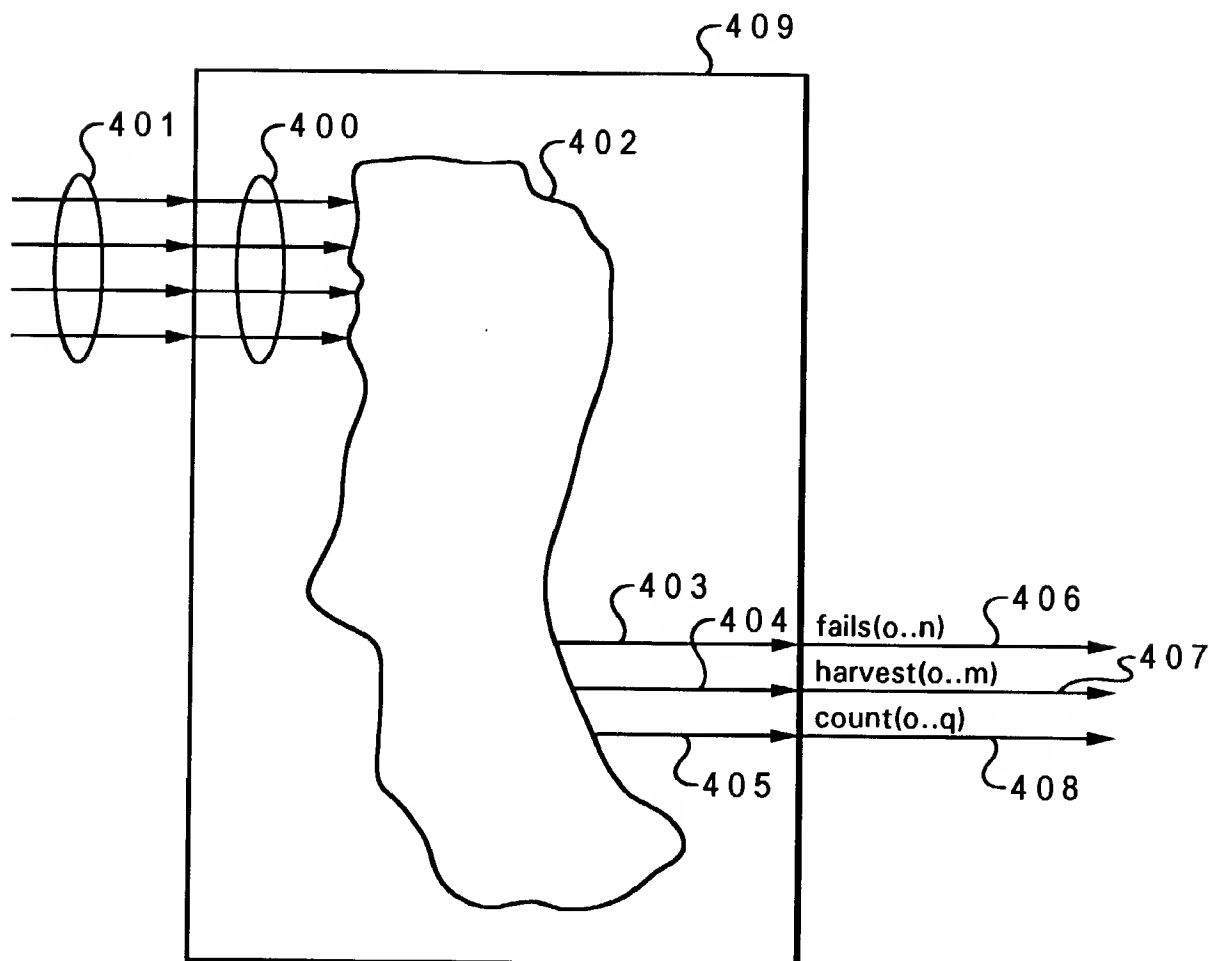


Fig. 3D



*Fig. 4A*

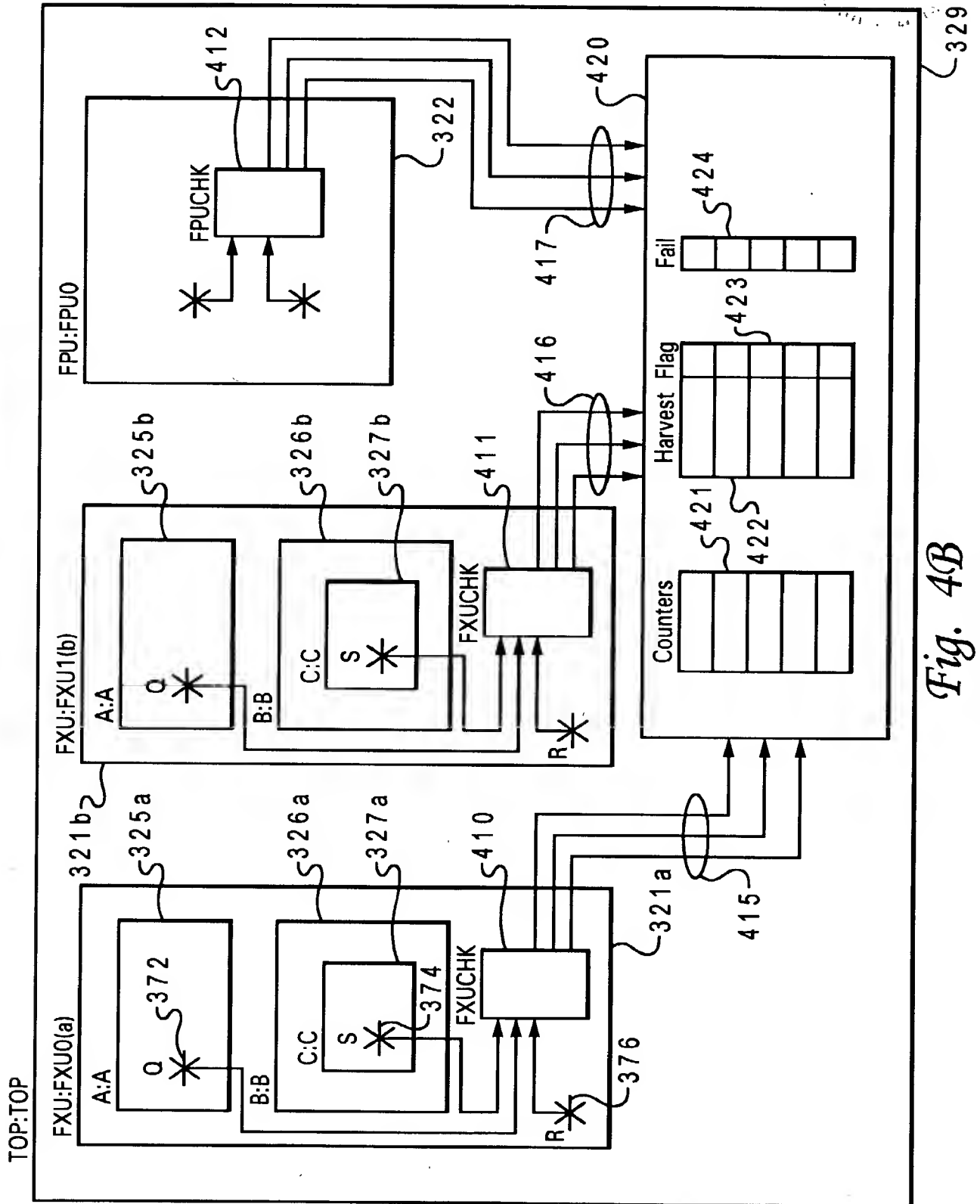


Fig. 4B



AUS920000861US1  
Gabele, et al.  
Fail Thresholding In A Batch Simulation Farm Network

9/62

ENTITY FXUCHK IS

```

PORT(  S_IN      :  IN std_ulogic;
        Q_IN      :  IN std_ulogic;
        R_IN      :  IN std_ulogic;
        clock     :  IN std_ulogic;
        fails     :  OUT std_ulogic_vector(0 to 1);
        counts    :  OUT std_ulogic_vector(0 to 2);
        harvests  :  OUT std_ulogic_vector(0 to 1);
);

```

4 5 2 { --!! BEGIN  
--!! Design Entity: FXU;

4 5 3 { --!! Inputs  
--!! S\_IN => B.C.S;  
--!! Q\_IN => A.Q;  
--!! R\_IN => R;  
--!! CLOCK => clock;  
--!! End Inputs

4 5 4 { --!! Fail Outputs;  
--!! 0 : "Fail message for failure event 0";  
--!! 1 : "Fail message for failure event 1";  
--!! End Fail Outputs;

4 5 5 { --!! Count Outputs;  
--!! 0 : <event0> clock;  
--!! 1 : <event1> clock;  
--!! 2 : <event2> clock;  
--!! End Count Outputs;

4 5 6 { --!! Harvest Outputs;  
--!! 0 : "Message for harvest event 0";  
--!! 1 : "Message for harvest event 1";  
--!! End Harvest Outputs;

4 5 7 { --!! End;

ARCHITECTURE example of FXUCHK IS

BEGIN

... HDL code for entity body section ...

END;

4 5 0

4 5 1

4 4 0

4 5 8

*Fig. 4C*



*Fig. 4D*

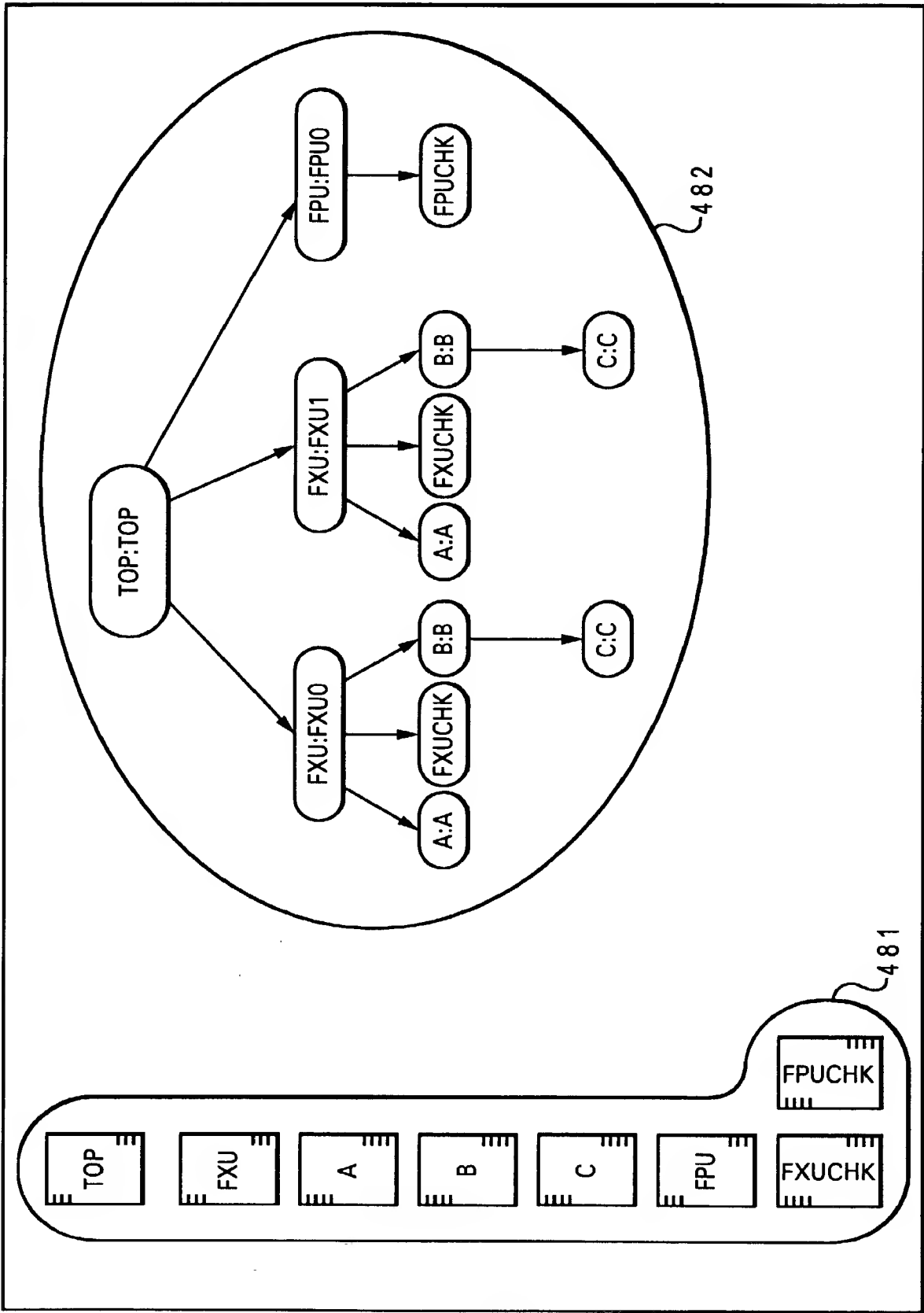
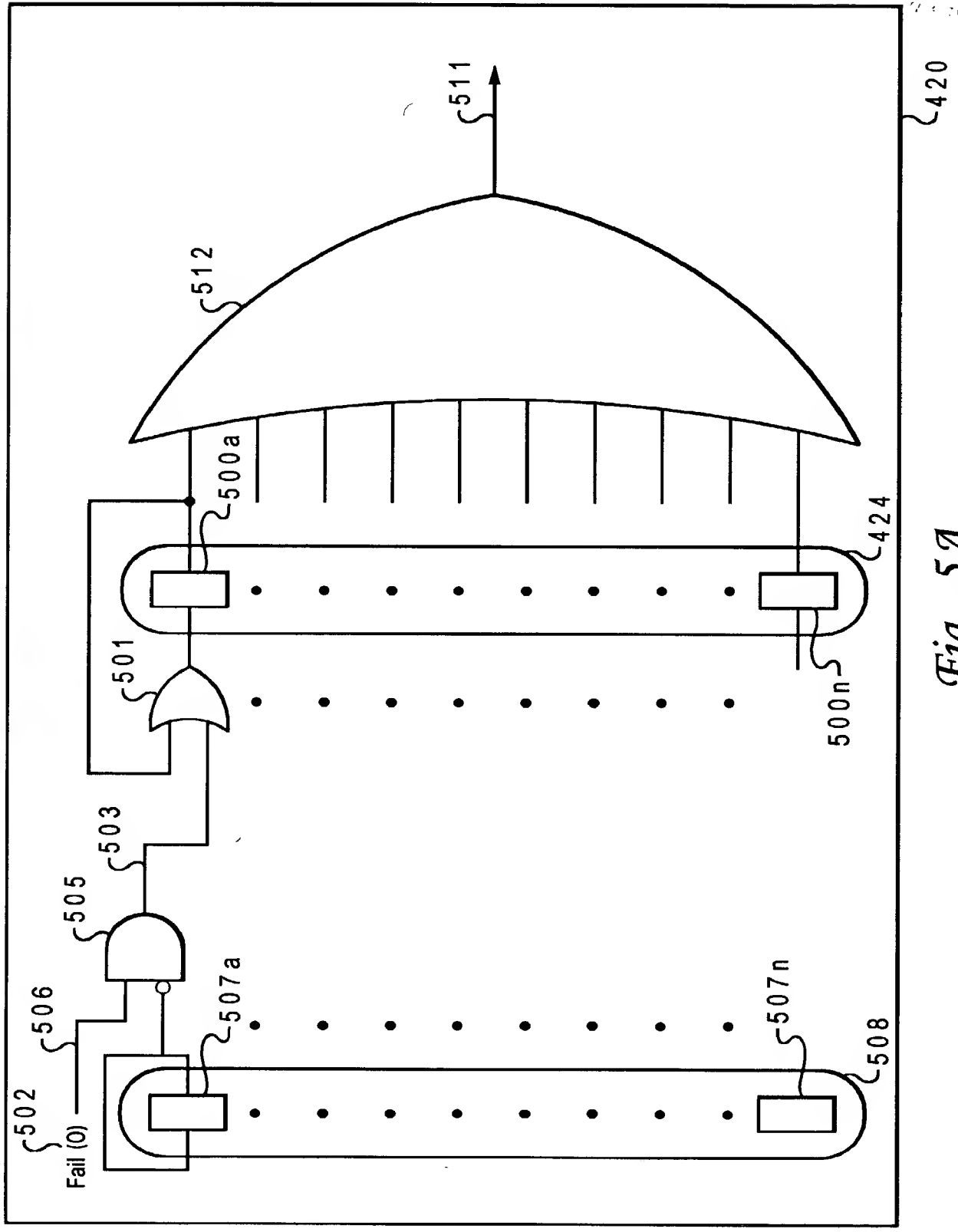


Fig. 4E



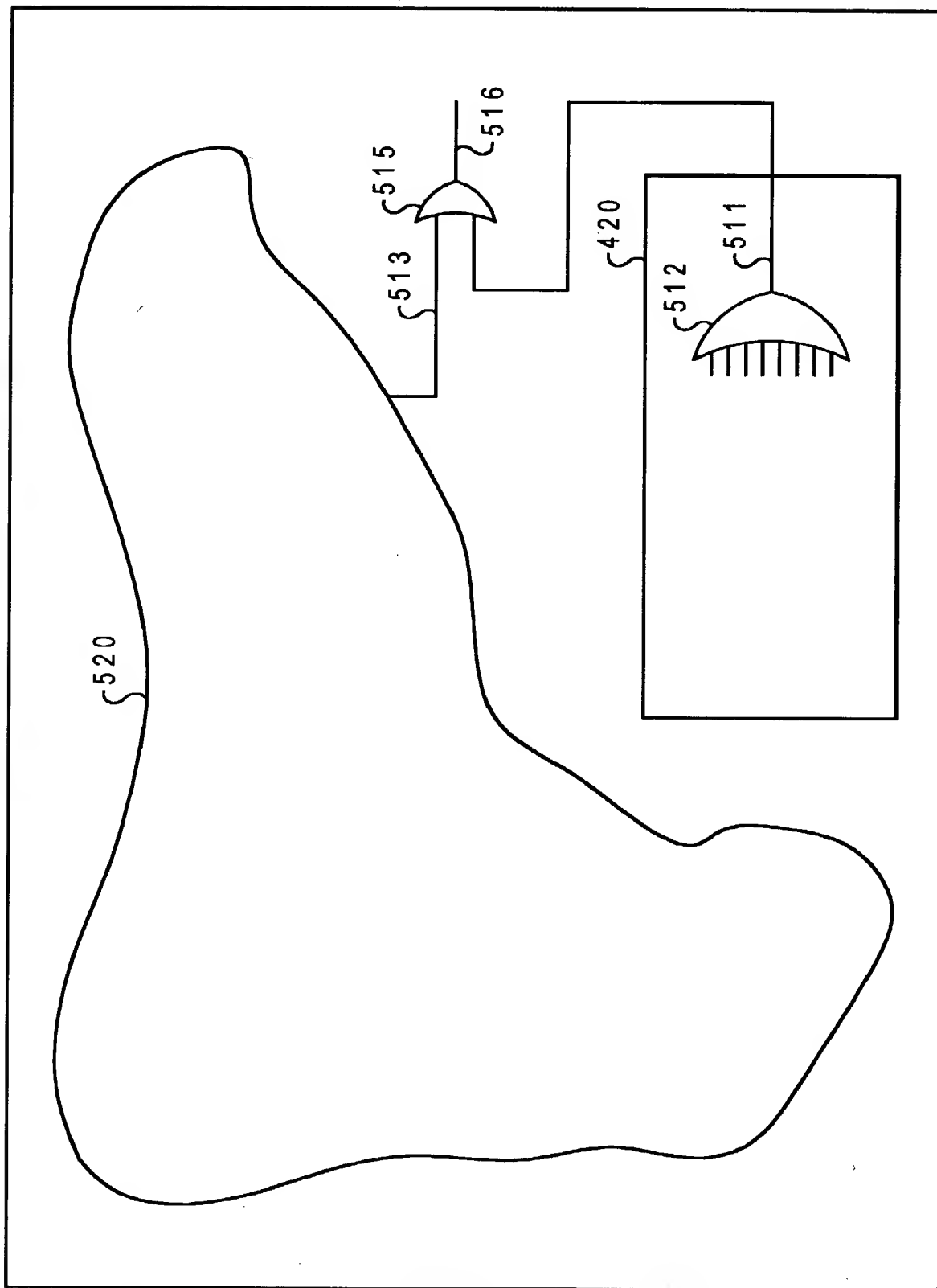
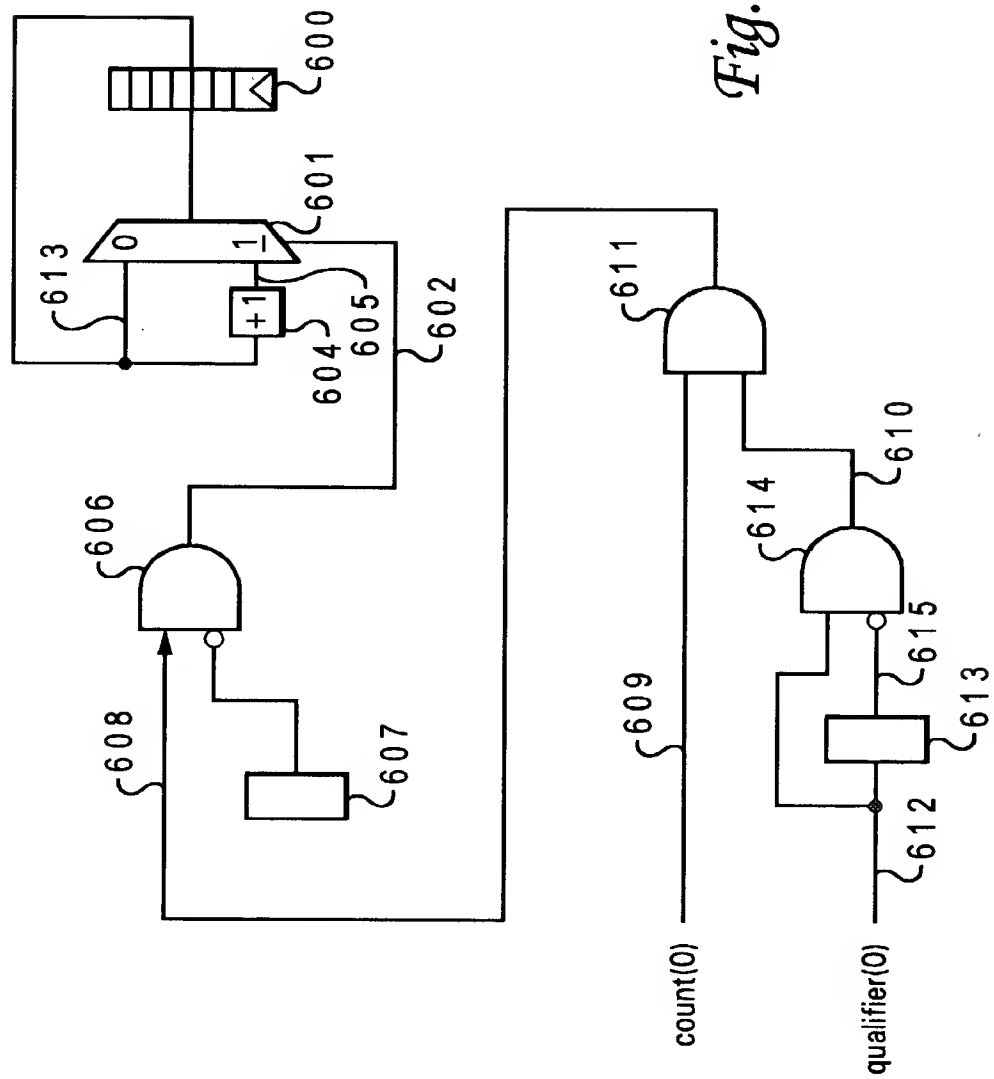


Fig. 5B



*Fig. 6A*

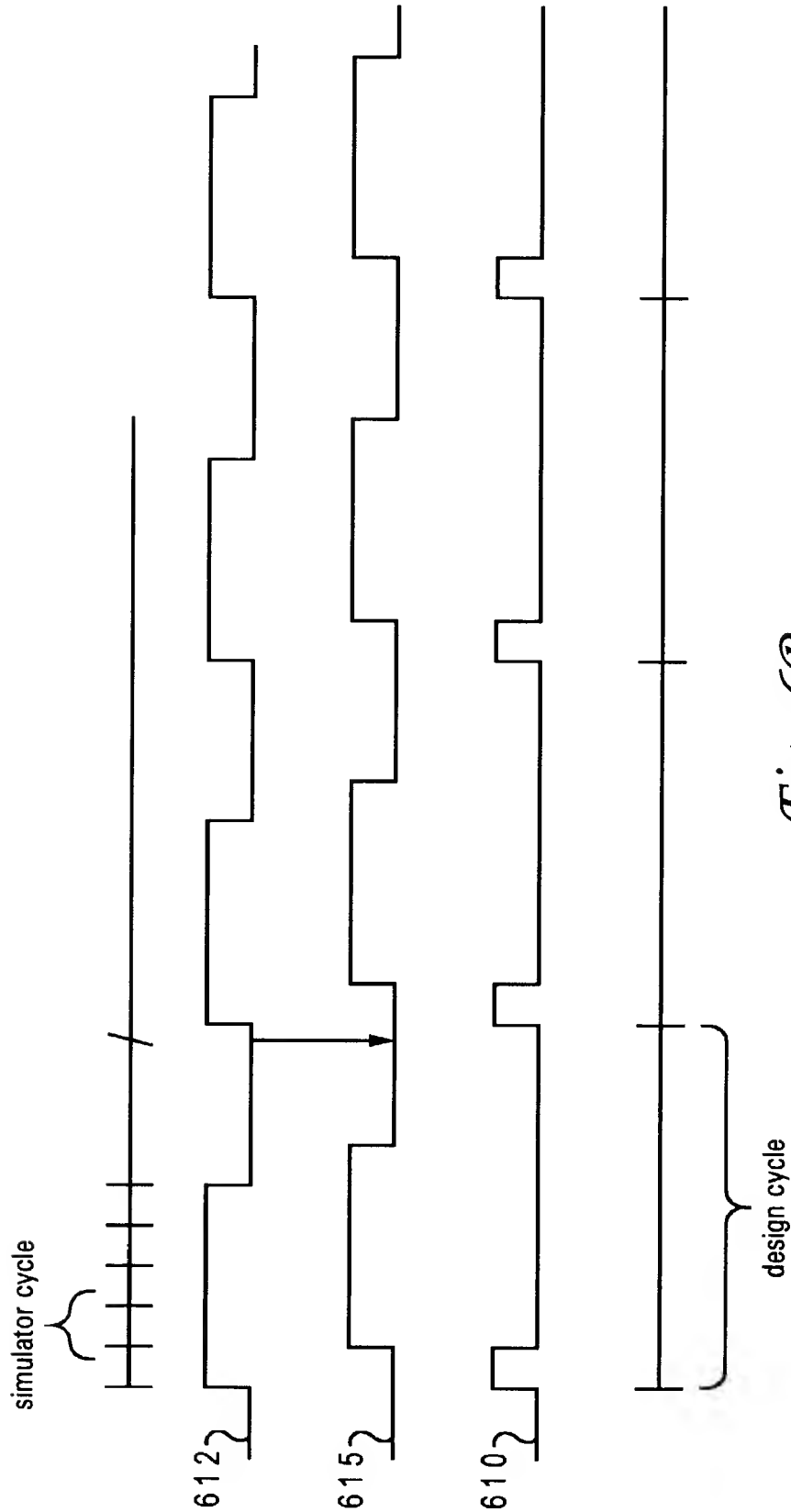


Fig. 6B

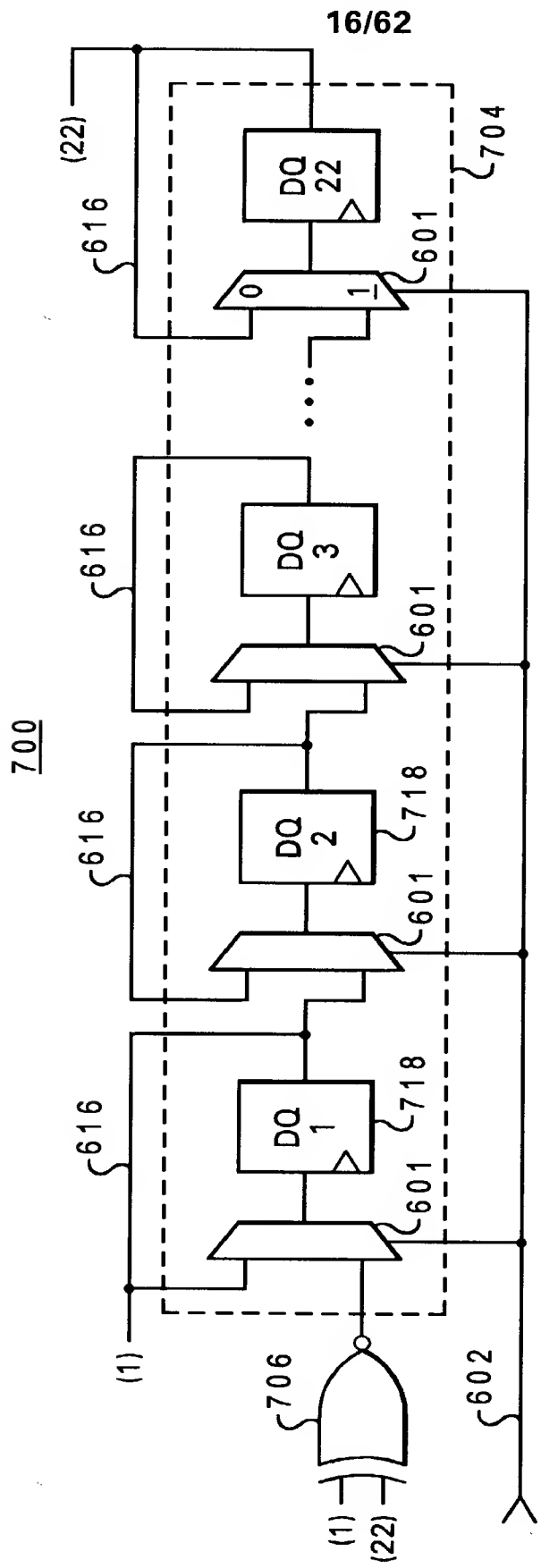
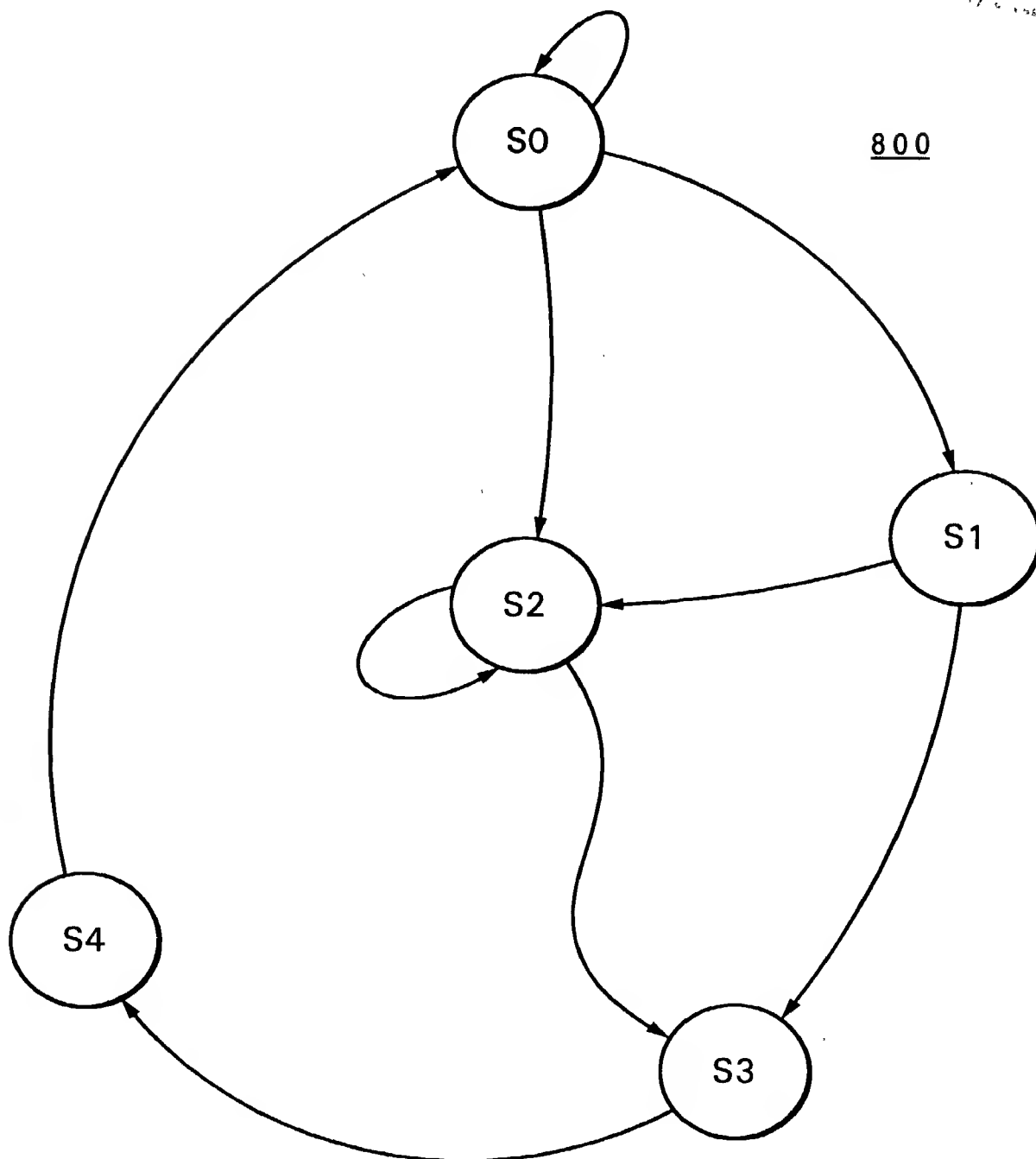


Fig. 7



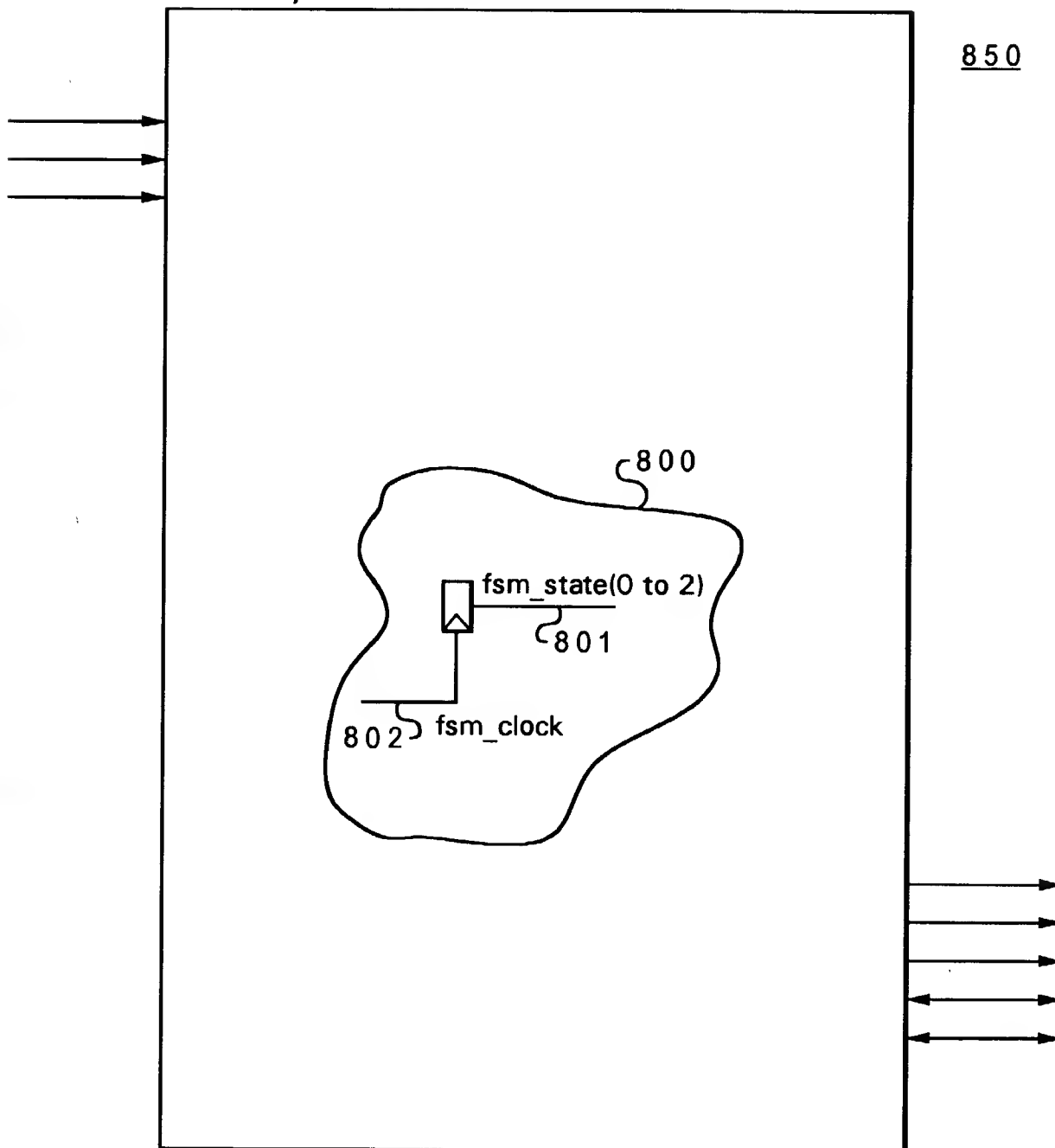
800



*Fig. 8A*  
*Prior Art*

entity FSM : FSM

850



*Fig. 8B*

*Prior Art*

AUS920000861US1  
Gabele, et al.  
Fail Thresholding In A Batch Simulation Farm Network

19/62

ENTITY FSM IS

PORT(  
....ports for entity fsm....  
);

ARCHITECTURE FSM OF FSM IS

BEGIN

... HDL code for FSM and rest of the entity ...

fsm\_state(0 to 2) <= ... Signal 801 ...

853	{	--!! Embedded FSM : examplefsm;	}	852	}	860
859	{	--!! clock : (fsm_clock);				
854	{	--!! state_vector : (fsm_state(0 to 2));				
855	{	--!! states : (S0, S1, S2, S3, S4);				
856	{	--!! state_encoding : ('000', '001', '010', '011', '100');				
857	{	--!! arcs : (S0 => S0, S0 => S1, S0 => S2, --!! (S1 => S2, S1 => S3, S2 => S2, --!! (S2 => S3, S3 => S4, S4 => S0);				
858	{	--!! End FSM;				

END;

*Fig. 8C*

20/62



entity FSM : FSM

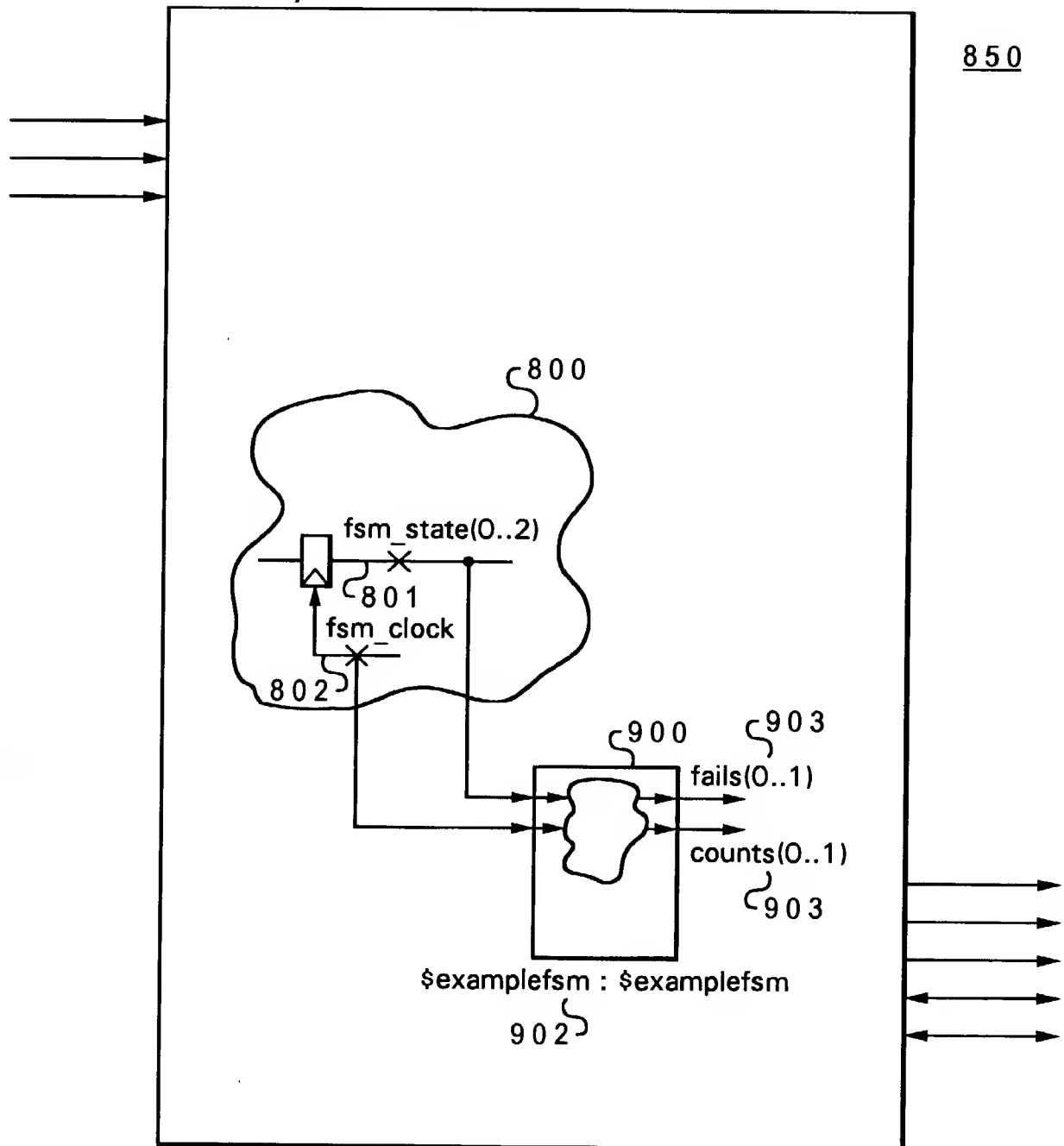
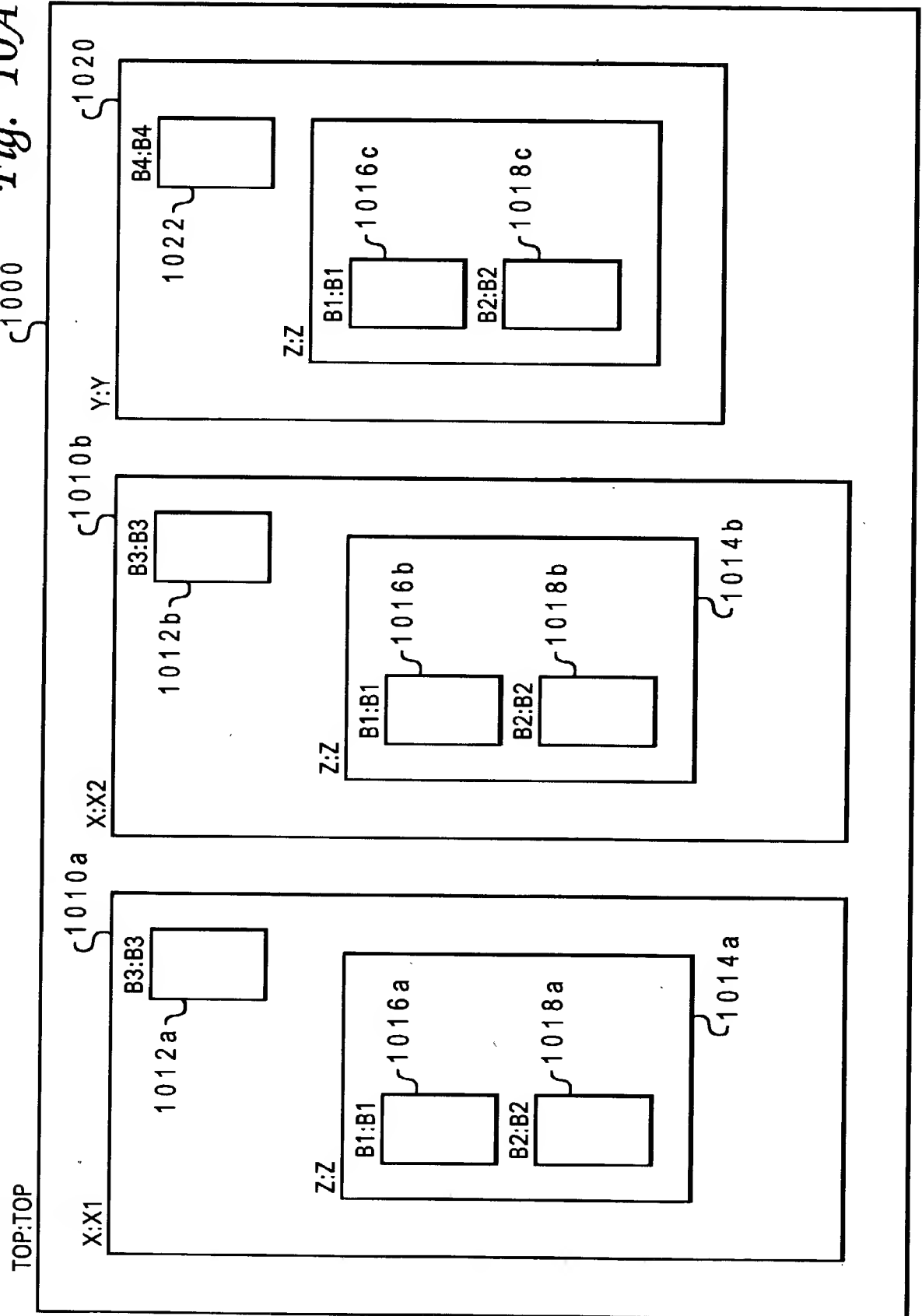


Fig. 9

Fig. 10A



AUS920000861US1

Gabele, et al.

Fail Thresholding In A Batch Simulation Farm Network

22/62

1030 1032 1034 1036  
 <instantiation identifier> . <instrumentation entity name> . <design entity name> . <eventname>

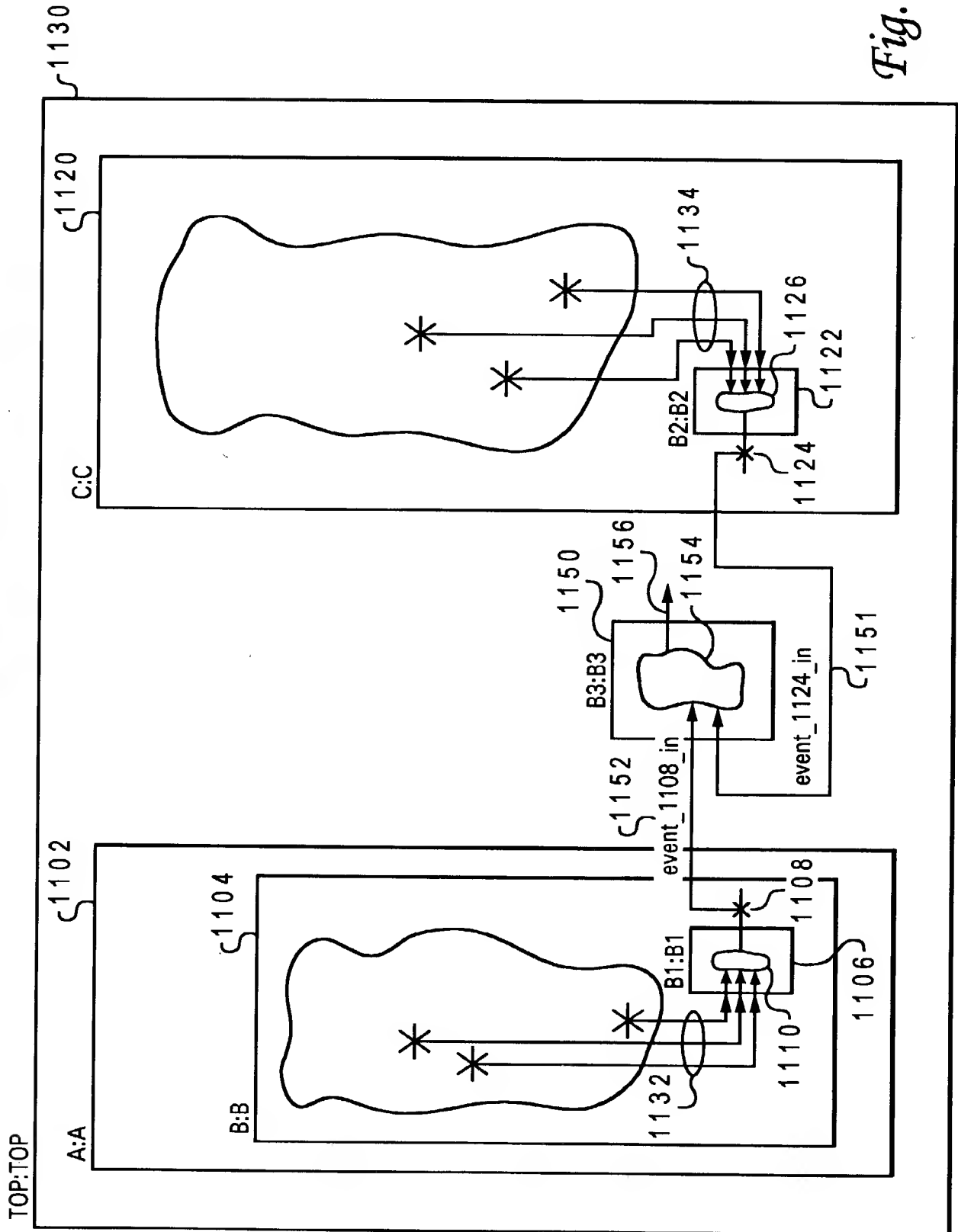
Fig. 10B

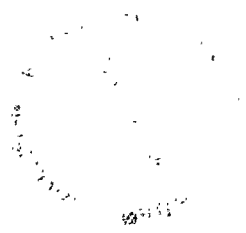
1030	1032	1034	1036
X1	B3	X	COUNT1
X1.Z	B1	Z	COUNT1
X1.Z	B2	Z	COUNT1
X2	B3	X	COUNT1
X2.Z	B1	Z	COUNT1
X2.Z	B2	Z	COUNT1
Y	B4	Y	COUNT1
Y.Z	B1	Z	COUNT1
Y.Z	B2	Z	COUNT1
			1040
			1041
			1042
			1043
			1044
			1045
			1046
			1047
			1048

Fig. 10C

1030 1034 1036  
 <instantiation identifier> . <design entity name> . <eventname>

Fig. 10D





--!! Inputs  
--!! event\_1108\_in <= C.[B2.count.event\_1108];  
--!! event\_1124\_in <= A.B.[B1.count.event\_1124];  
--!! End Inputs

Diagrammatic annotations for Fig. 11B:  
- A bracket labeled 1163 spans the first two lines of code.  
- A bracket labeled 1165 spans the right-hand side of the first two lines of code.  
- A bracket labeled 1164 spans the first two lines of code.  
- A bracket labeled 1166 spans the right-hand side of the first two lines of code.  
- A wavy line labeled 1161 is next to the first line of code.  
- A wavy line labeled 1162 is next to the second line of code.

*Fig. 11B*

--!! Inputs  
--!! event\_1108\_in <= C.[count.event\_1108];  
--!! event\_1124\_in <= B.[count.event\_1124];  
--!! End Inputs

Diagrammatic annotations for Fig. 11C:  
- A wavy line labeled 1171 is next to the first line of code.  
- A wavy line labeled 1172 is next to the second line of code.

*Fig. 11C*

20090207 29445660





```

ENTITY X IS
    PORT(
        :
        :
        :
    );

    ARCHITECTURE example of X IS
    BEGIN
        .
        .
        .
        ... HDL code for X ...
        .
        .
        .

1 2 2 1 { Y:Y
        PORT MAP(
            :
            :
            );

1 2 2 2 { A <= ....
        B <= ....
        C <= ....

1 2 2 3 { --!! [count, countname0, clock] <= Y.P;
        --!! Q <= Y. [B1.count.count1] AND A;
        --!! [fail, failname0, "fail,msg"] <= Q XOR B;
        --!! [harvest, harvestname0, "harvest,msg"] <= B AND C;

        END;
    
```

1 2 2 0

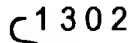
1 2 3 0

1 2 3 2

1 2 3 4

1 2 3 6

*Fig. 12B*



*Fig. 13A*

FOO:FOO

1300

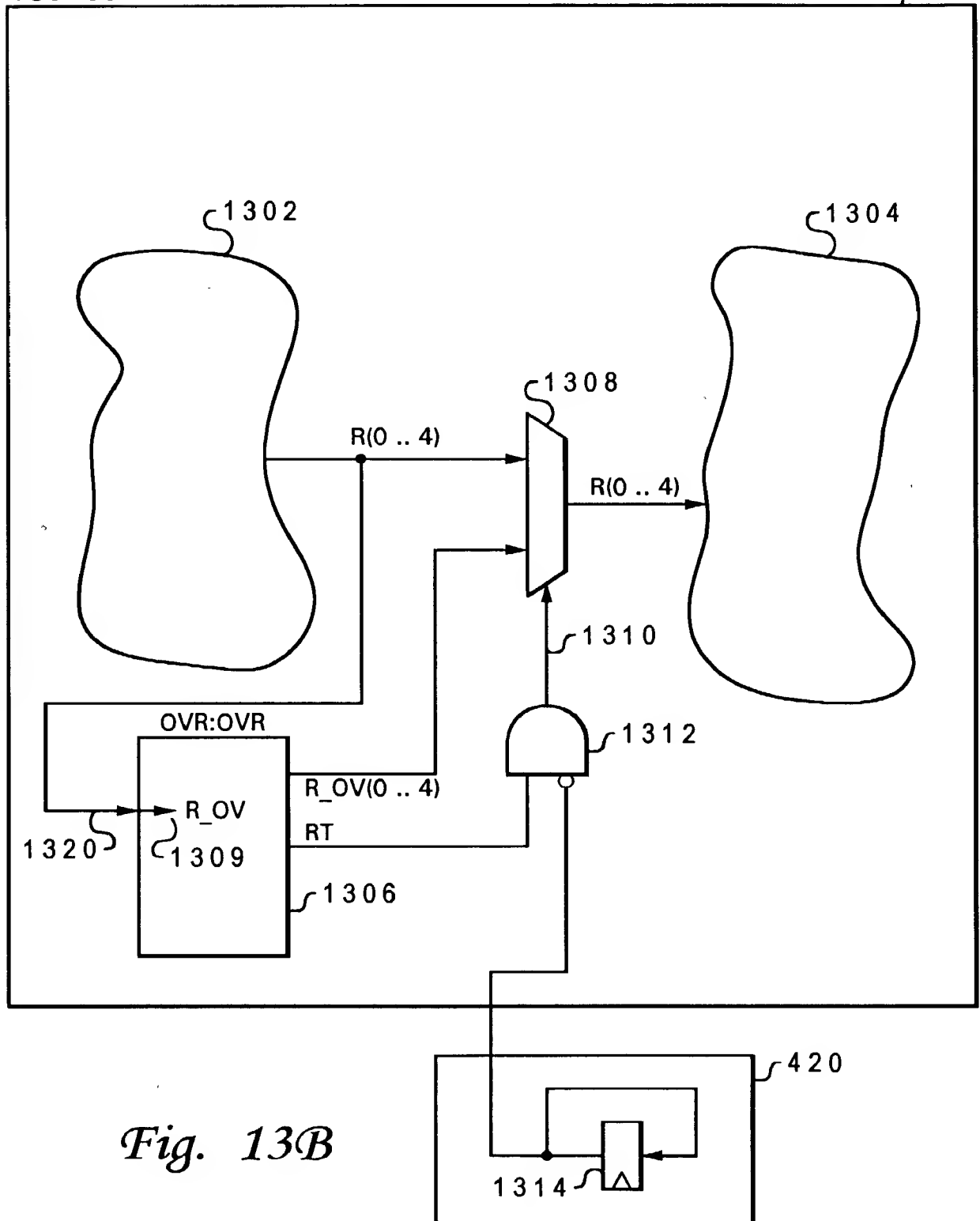


Fig. 13B

29/62

```

ENTITY OVR IS
    PORT( R_IN      : IN std_ulogic_vector(0 .. 4);
          .
          .
          ... other ports as required ...
          .
          R_OV      : OUT std_ulogic_vector(0 .. 4);
          RT        : OUT std_ulogic
    );

--!! BEGIN
--!! Design Entity: FOO;

--!! Inputs (0 to 4)
--!! R_IN = > {R(0 .. 4)};
--!! :
... other ports as needed ...
--!! :
--!! End Inputs

--!! Outputs
--!! <R_OVERRIDE> : R_OV(0 .. 4) = > R(0 .. 4) [RT];
--!! End Outputs

--!! End

ARCHITECTURE example of OVR IS
    BEGIN
        ... HDL code for entity body section ...
    END;
    
```

Diagram annotations (brackets and labels):

- 1364: Bracket for the first PORT declaration.
- 1362: Bracket for the R\_OV and RT output declarations.
- 1363: Bracket for the output declarations.
- 1360: Bracket for the R\_IN input declaration.
- 1361: Bracket for the output declaration in the ARCHITECTURE.
- 1356: Bracket for the output declaration in the ARCHITECTURE.
- 1351: Bracket for the output declaration in the ARCHITECTURE.
- 1358: Bracket for the HDL code in the ARCHITECTURE.
- 1340: Large bracket on the right side, spanning from the first PORT declaration down to the end of the ARCHITECTURE.

*Fig. 13C*

30/62

ENTITY FOO IS

```
PORT(      :
          :
          :
          :
      );
```

## ARCHITECTURE example of FOO IS

**BEGIN**

$$R \leq \dots$$

```

1380 {
    --!! R_IN <= {R};
    --!!
    --!!
    --!! R_OV(0 to 4) <= .....;
    --!! RT <= .....;
    --!! [override, R_OVRRIDE, R(0 .. 4), RT] <= R_OV(0 to 4);
}

```

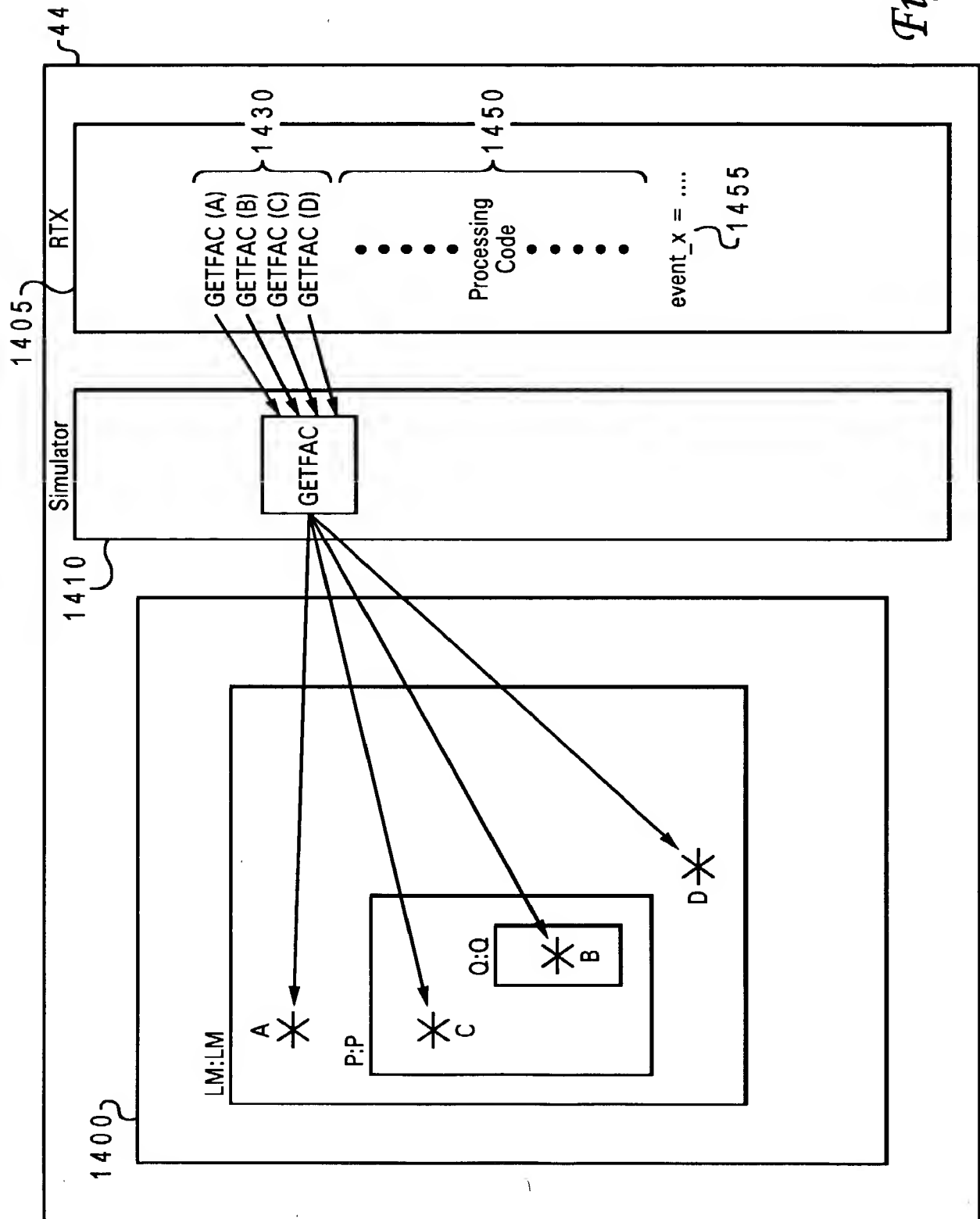
*Fig. 13D*

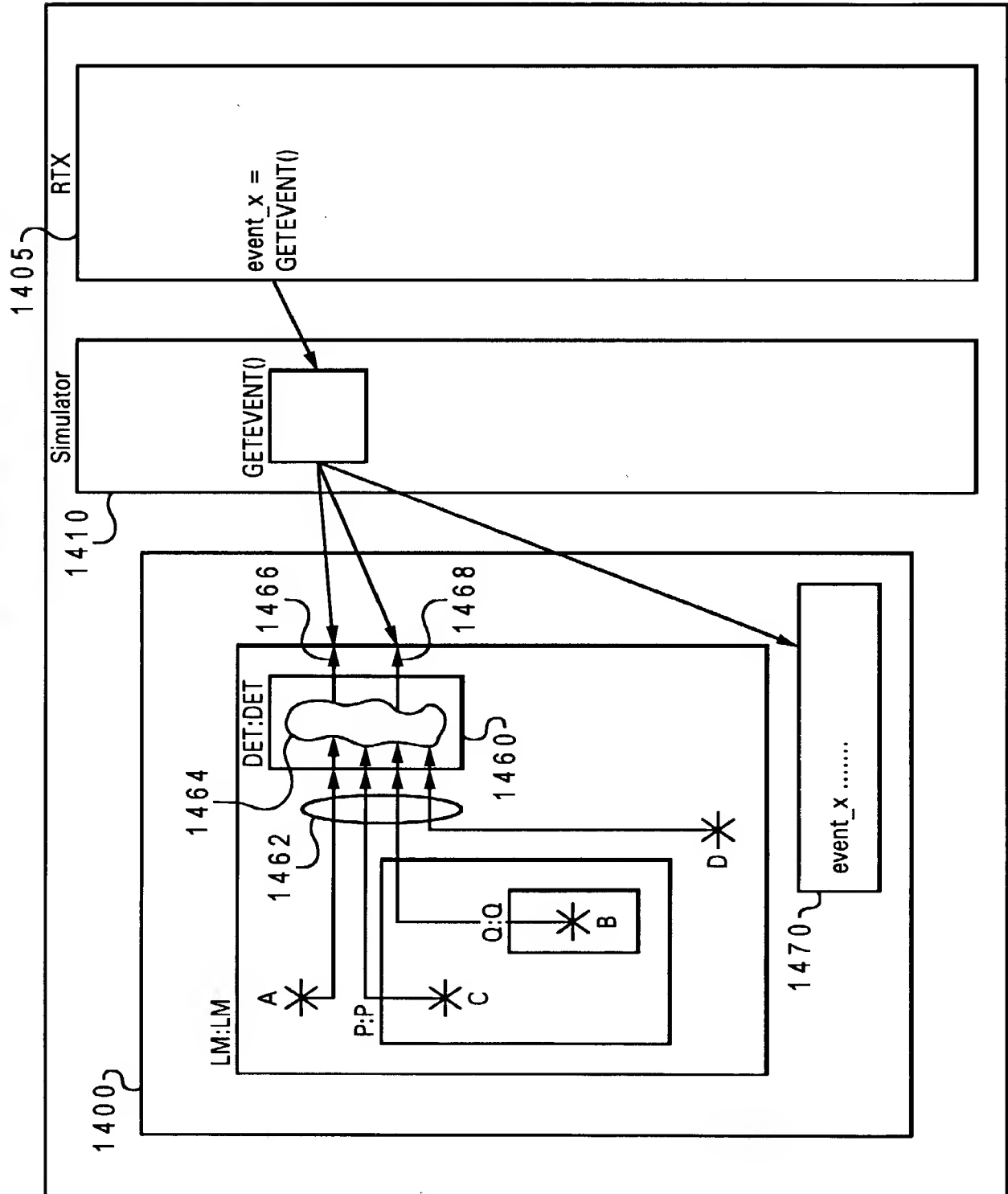
US 2003/0126031 A1

AUS920000861US1  
Gabele, et al.  
Fail Thresholding In A Batch Simulation Farm Network

31/62

Fig. 14A







```

ENTITY DET IS
    PORT(  A      :  IN std_ulogic;
           B      :  IN std_ulogic_vector(0 to 5);
           C      :  IN std_ulogic;
           D      :  IN std_ulogic;
           :
           :
           event_x :  OUT std_ulogic_vector(0 to 2);
           x_here  :  OUT std_ulogic;
    );

    --!! BEGIN
    --!! Design Entity: LM;

    --!! Inputs
    --!! A    =>  A;
    --!! B    =>  P.Q.B;
    --!! C    =>  P.C;
    --!! D    =>  D;
    --!! End Inputs

    --!! Detections
    --!! <event_x>:event_x(0 to 2) [x_here];
    --!! End Detections

    --!! End;

    ARCHITECTURE example of DET IS
    BEGIN
        ... HDL code ...

    END;

```

1491 {

1493 {

1495 {

1494 {

1480 {

1492 {

*Fig. 14C*

AUS920000861US1

Gabele, et al.

Fail Thresholding In A Batch Simulation Farm Network

34/62

1660

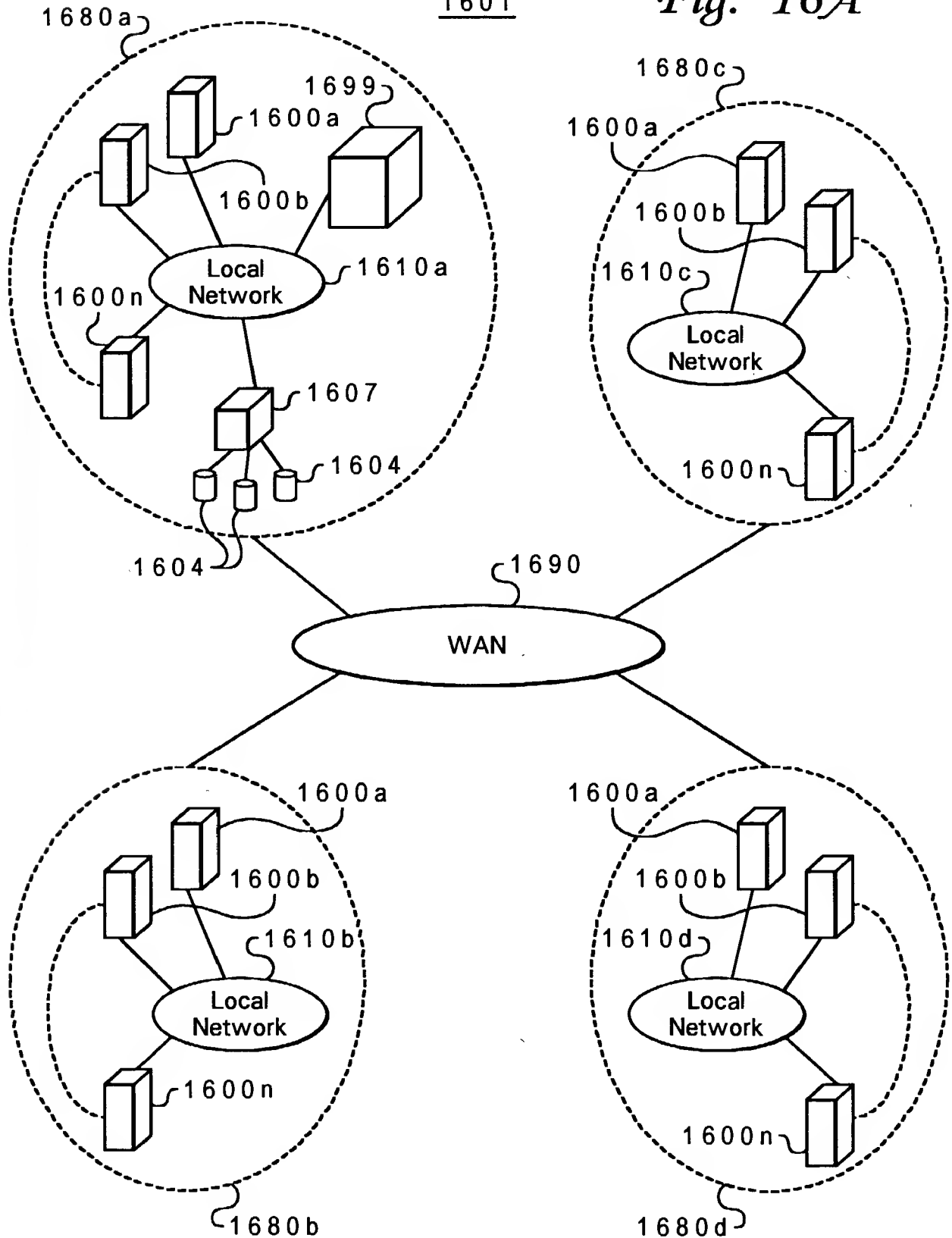
	1 6 6 1	1 6 6 2			
1 6 6 3	1:	X1	B3	X	COUNT1
	2:	X1.Z	B1	Z	COUNT1
	3:	X1.Z	B2	Z	COUNT1
	4:	X2	B3	X	COUNT1
	5:	X2.Z	B1	Z	COUNT1
	6:	X2.Z	B2	Z	COUNT1
	7:	Y	B4	Y	COUNT1
	8:	Y.Z	B1	Z	COUNT1
	9:	Y.Z	B2	Z	COUNT1

Fig. 15

35/62

1601

Fig. 16A



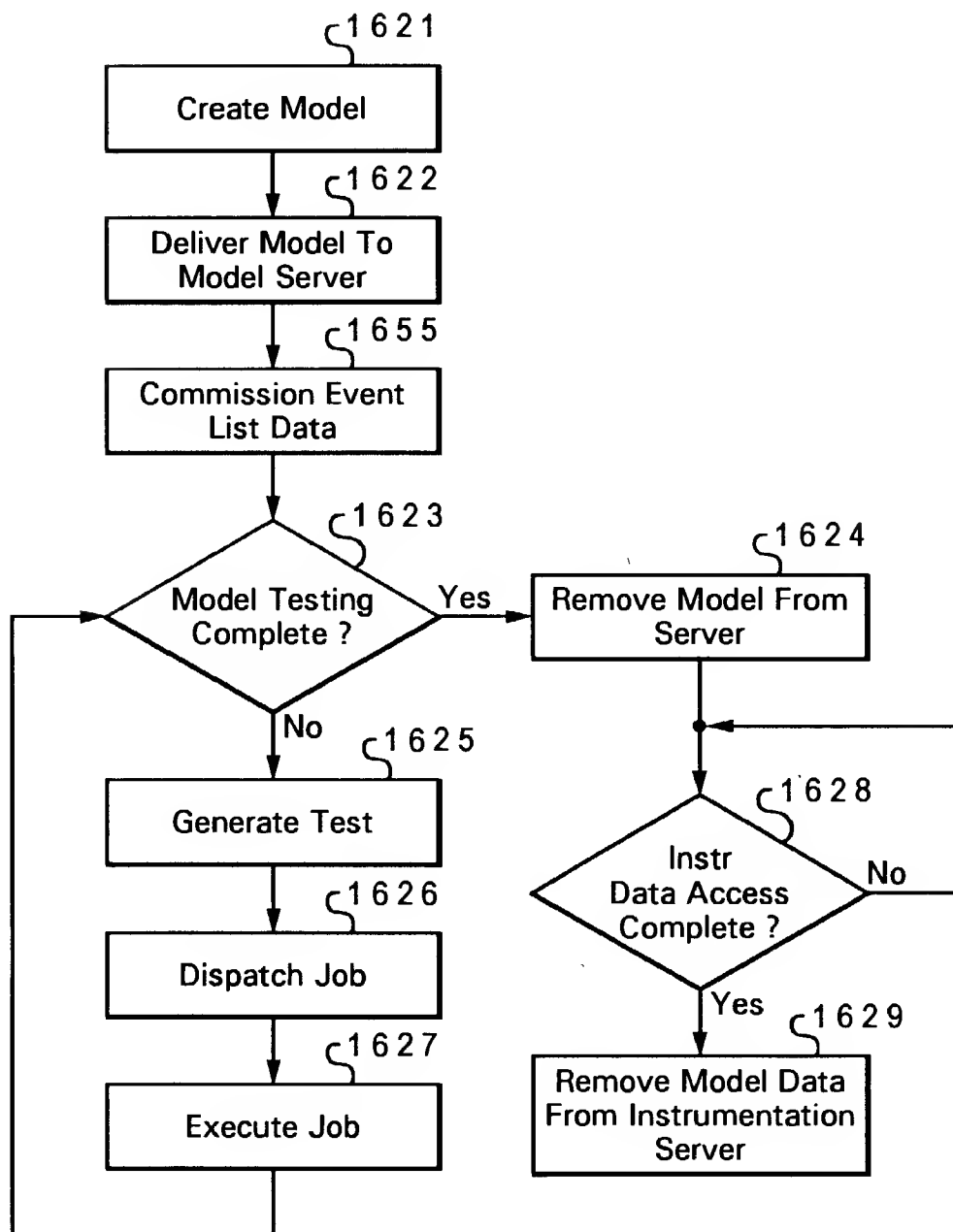
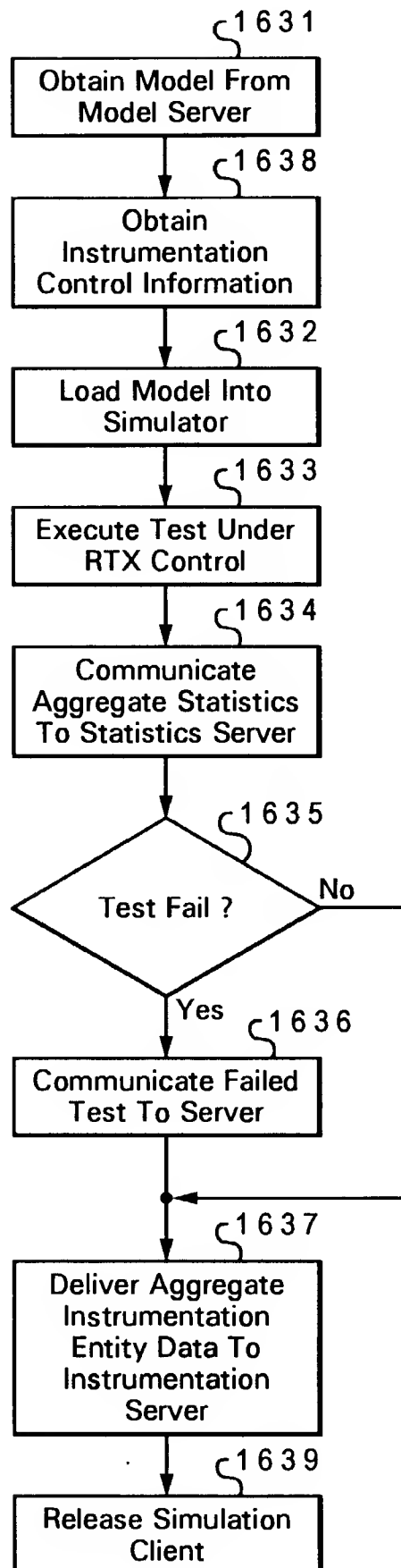


Fig. 16B



*Fig. 16C*

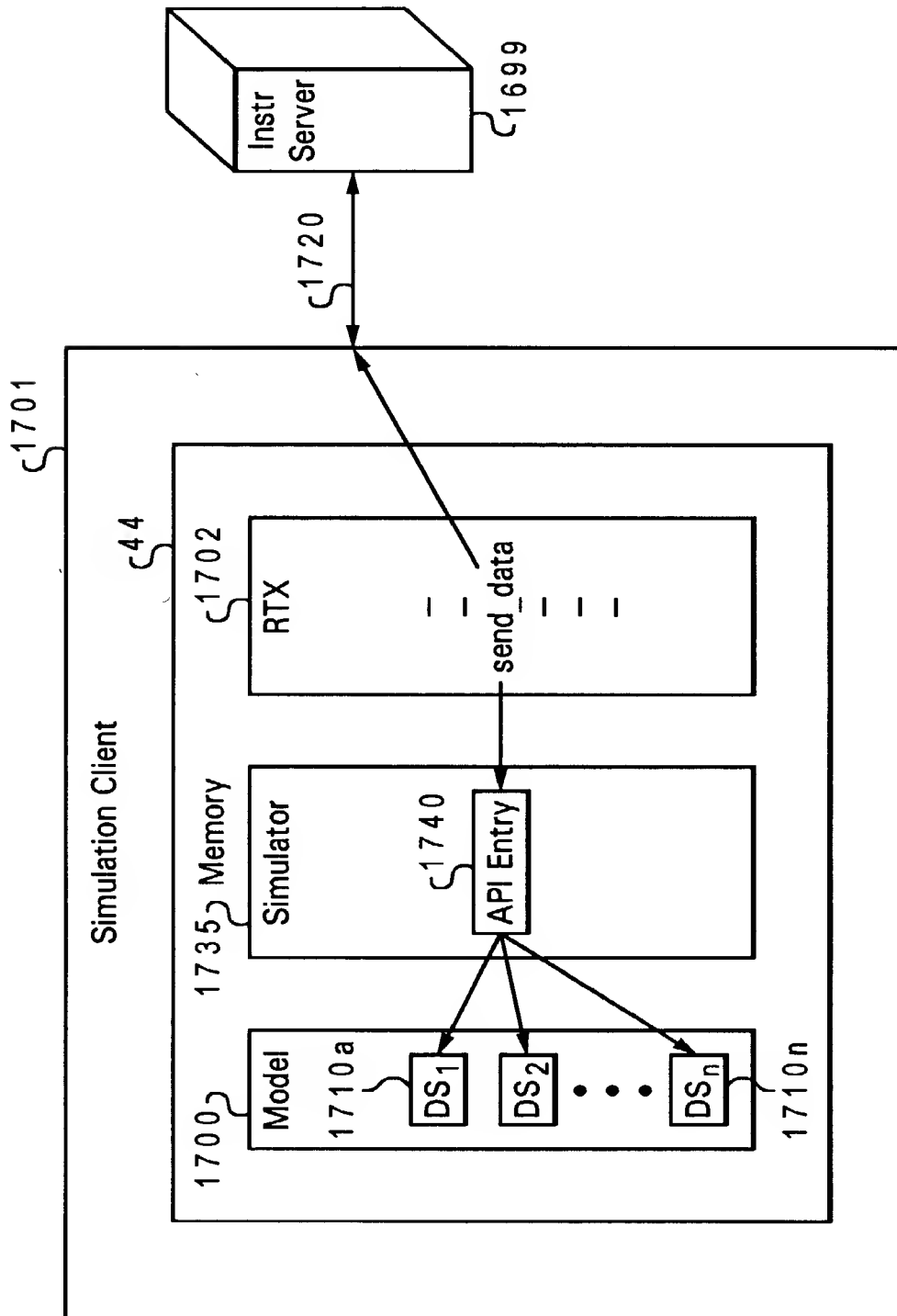


Fig. 17A



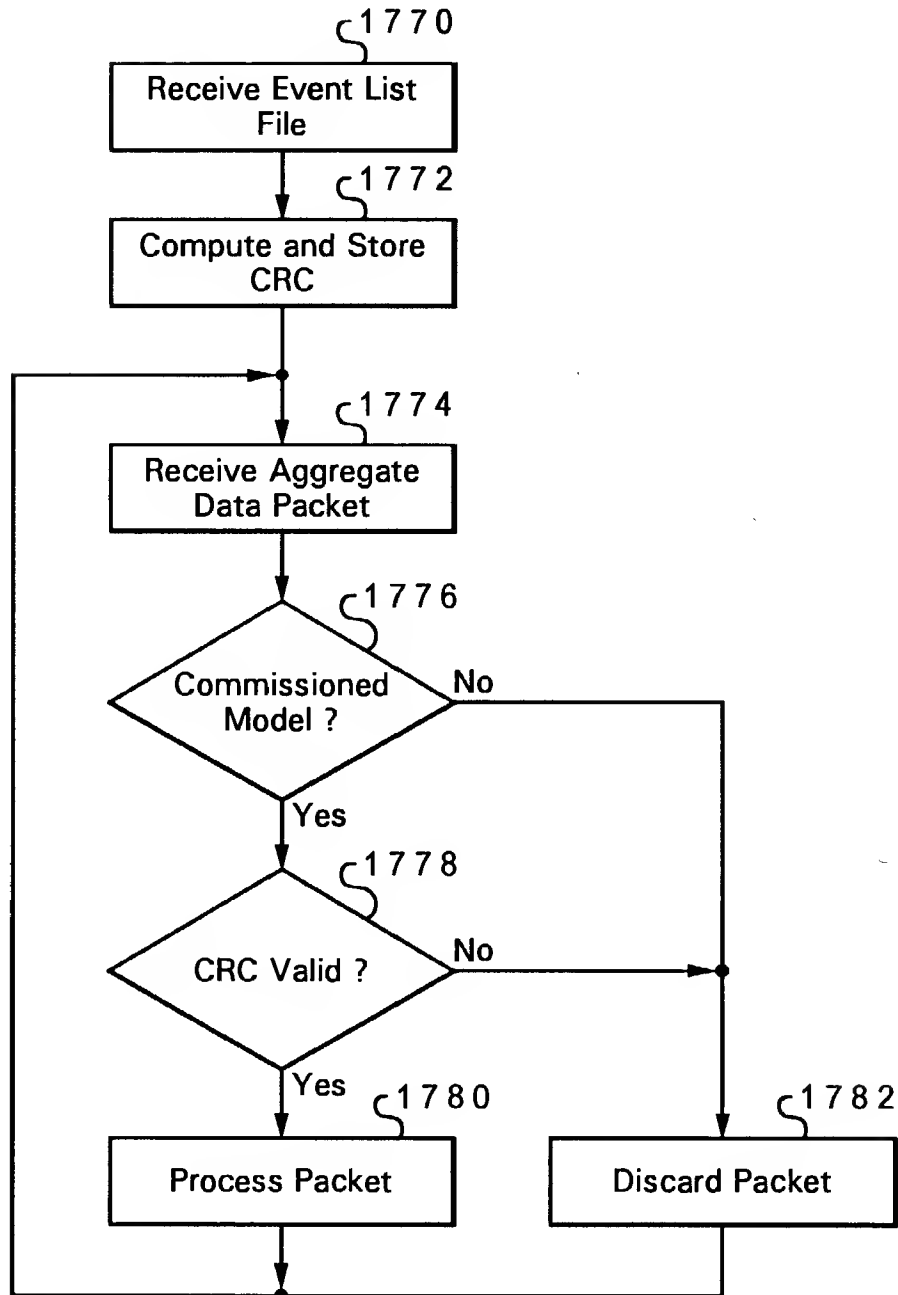


Fig. 17C



AUS920000861US1  
 Gabele, et al.  
 Fail Thresholding In A Batch Simulation Farm Network

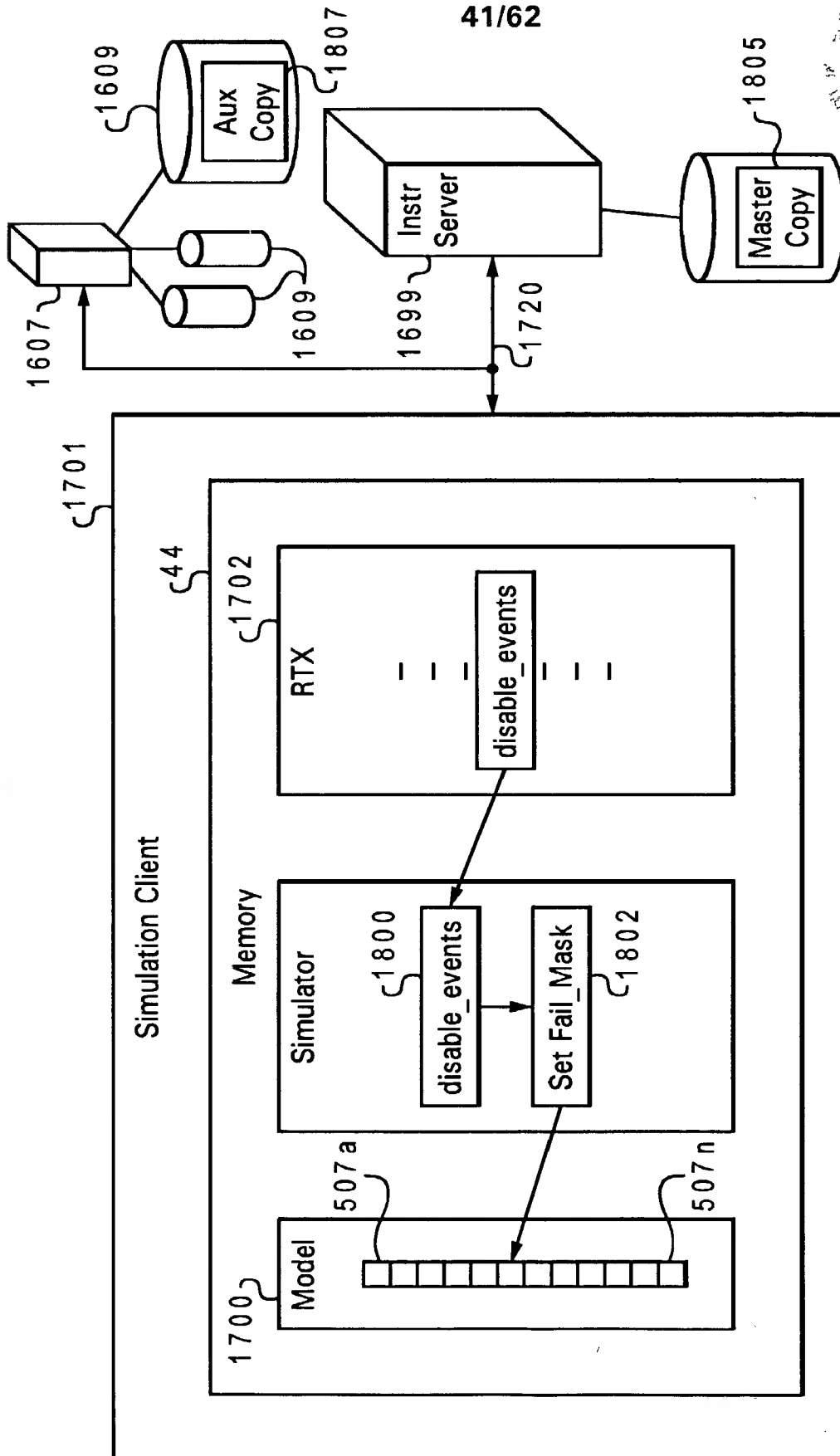
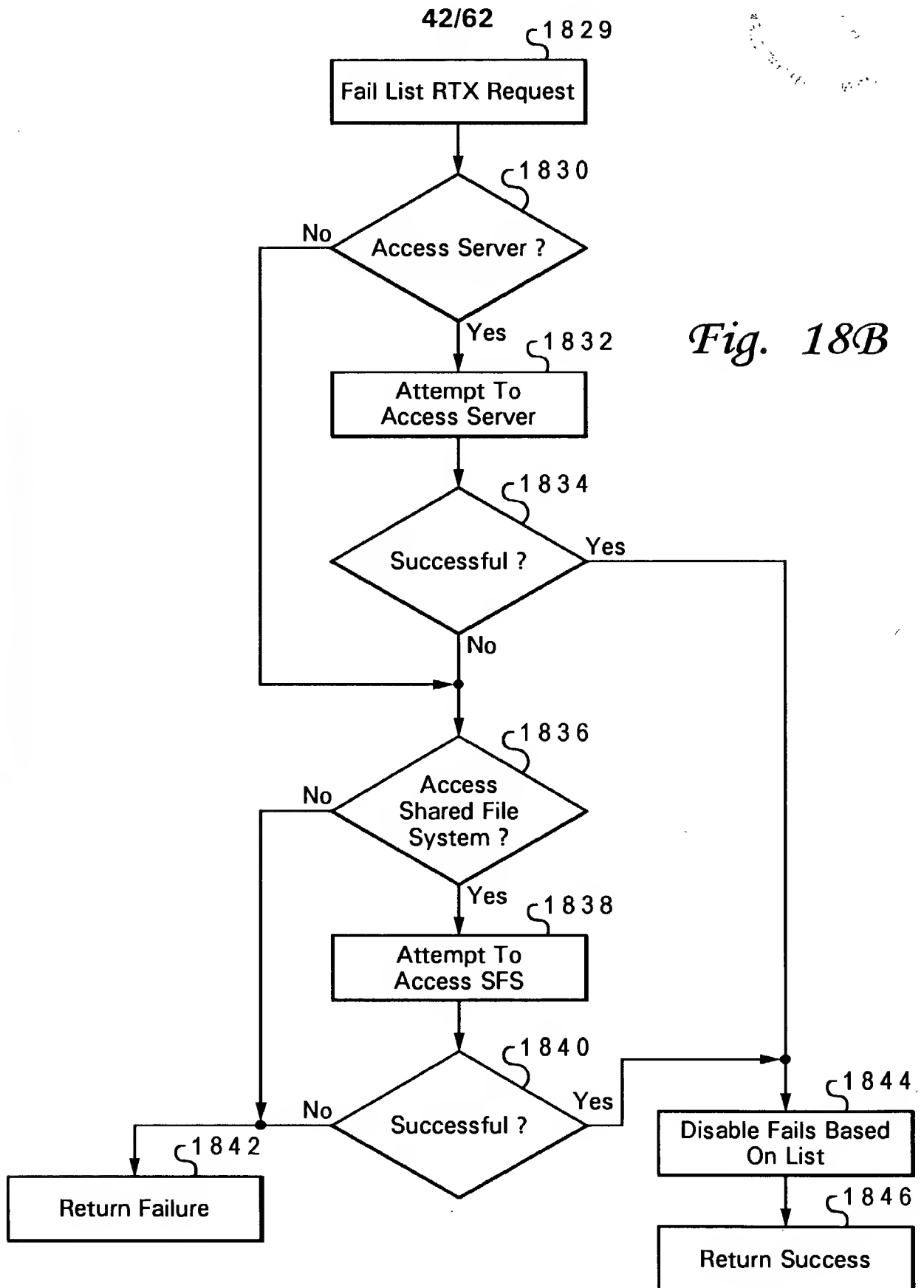


Fig. 18A



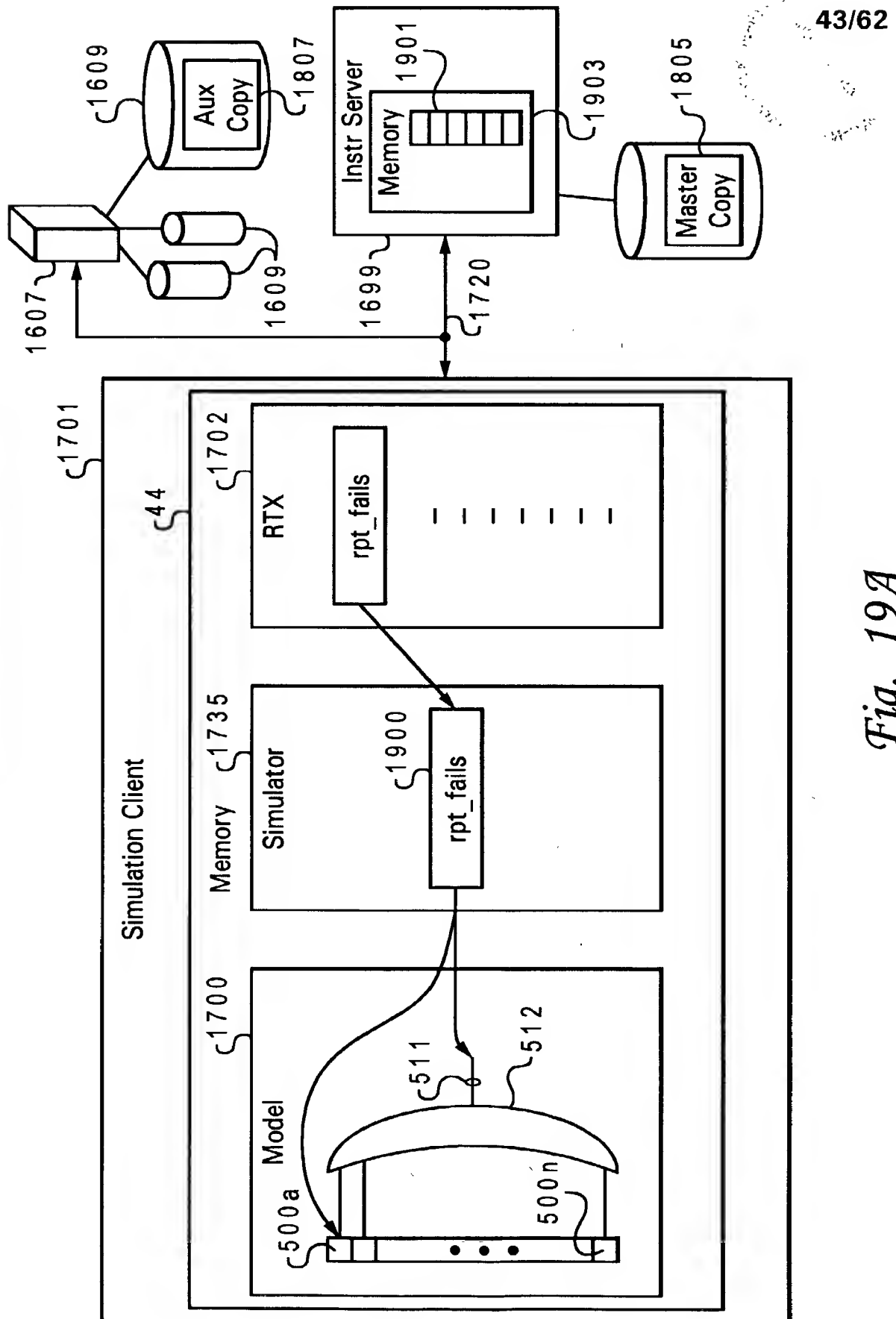
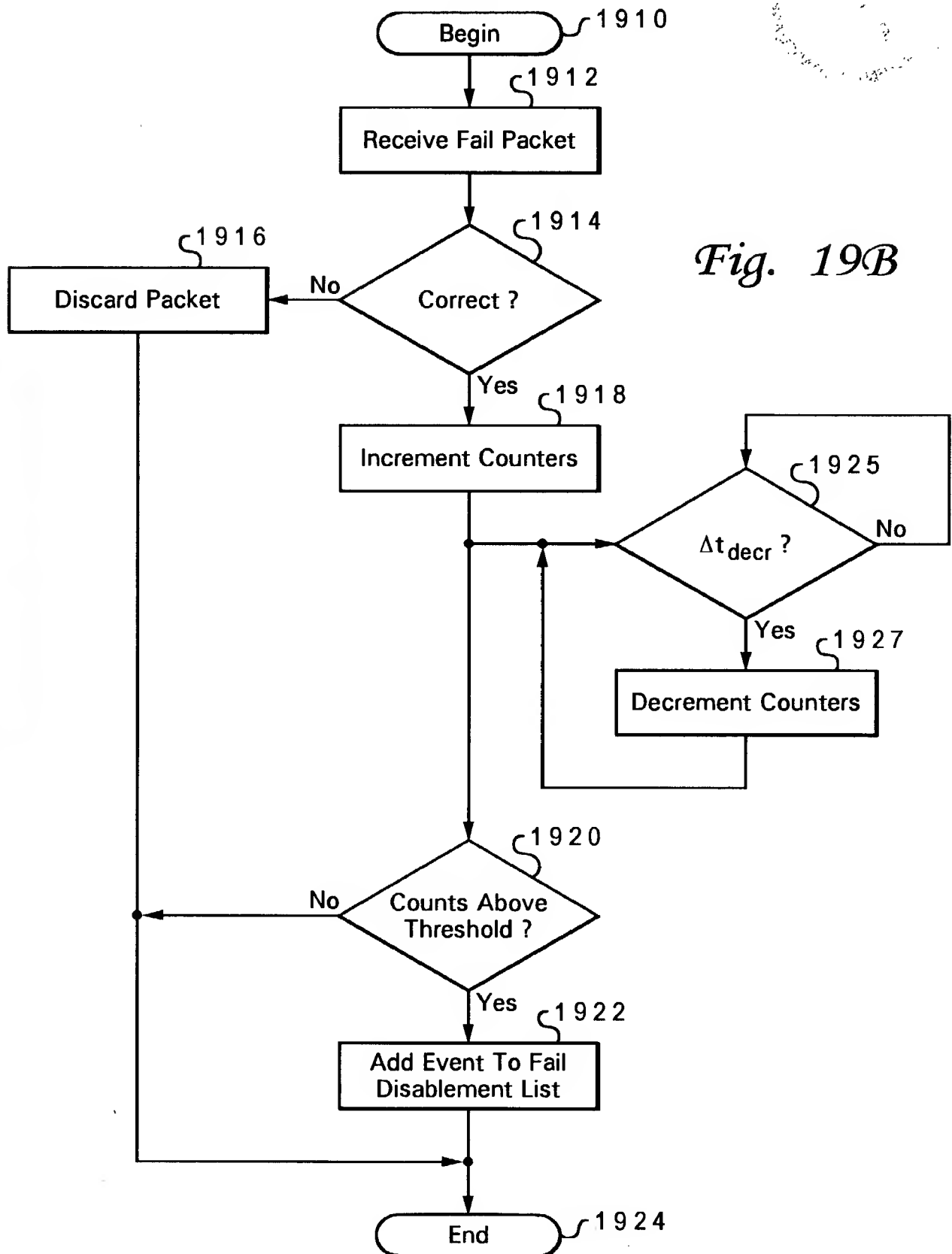


Fig. 19A

44/62



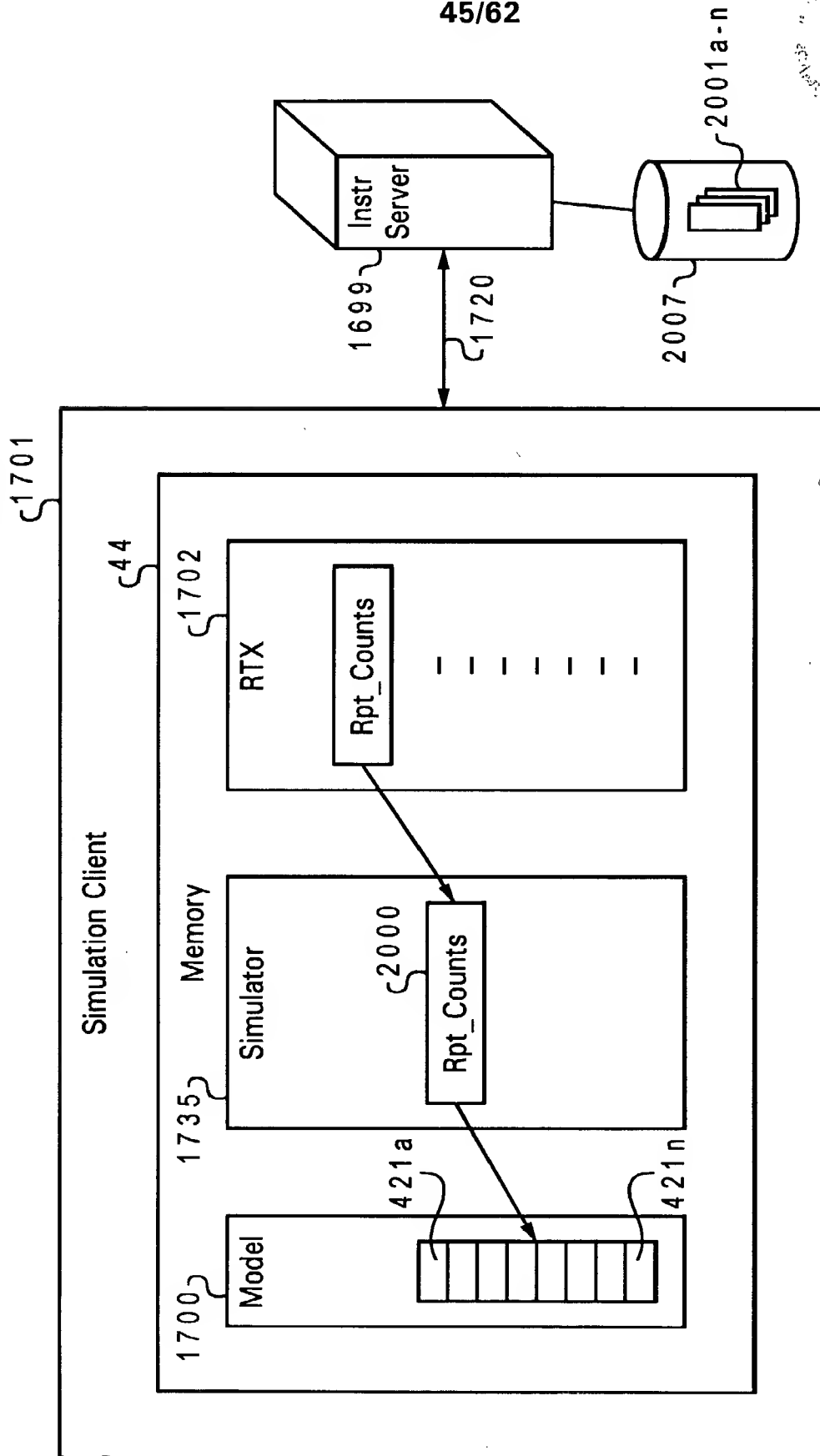


Fig. 20A

46/62

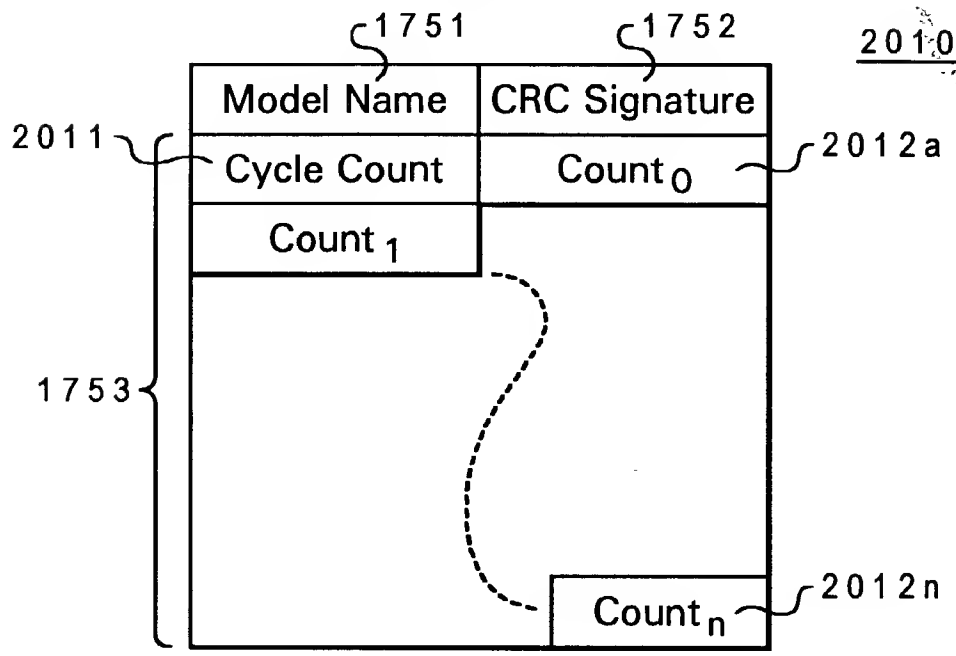


Fig. 20B

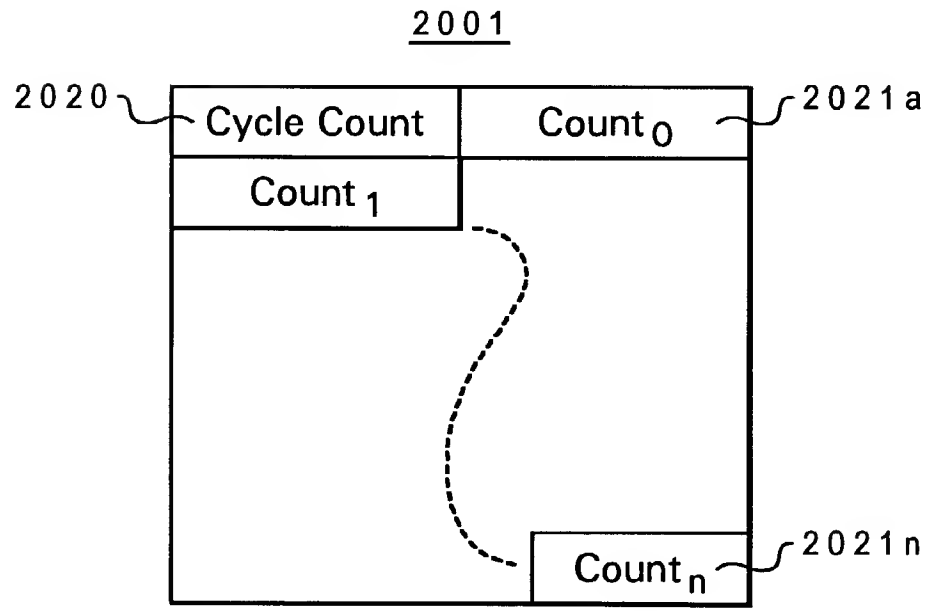


Fig. 20C

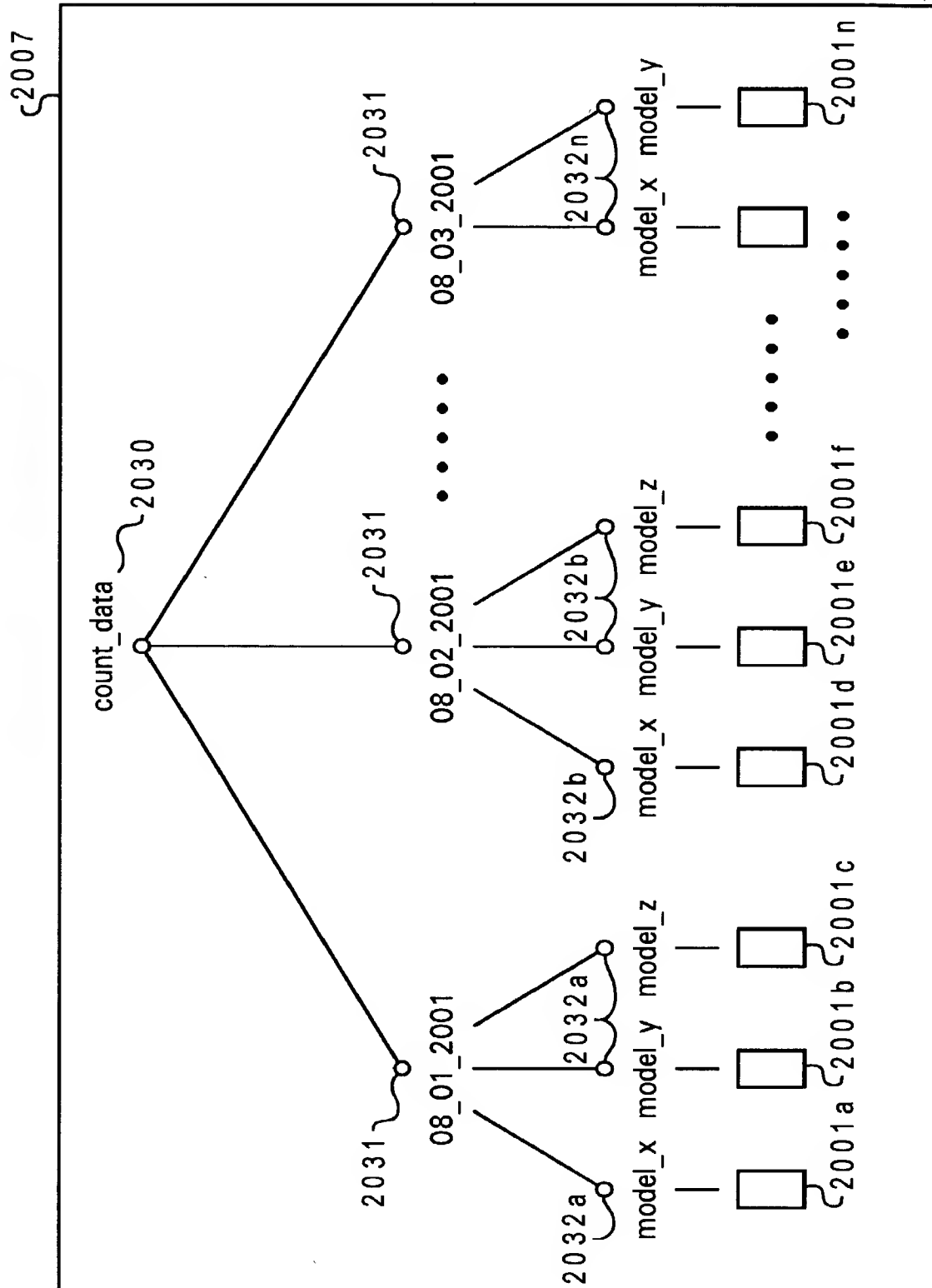
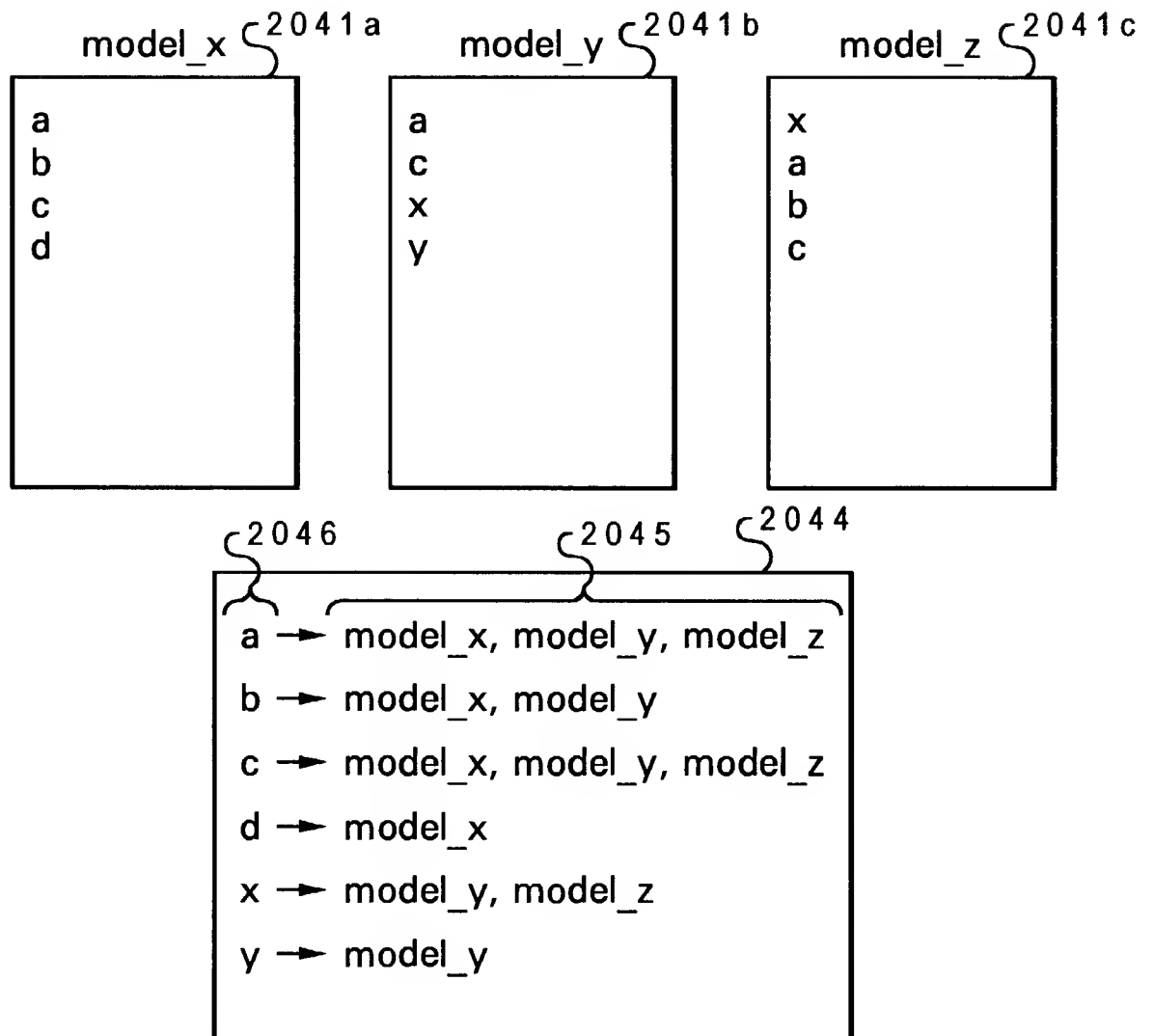


Fig. 20D



*Fig. 20E*



49/62

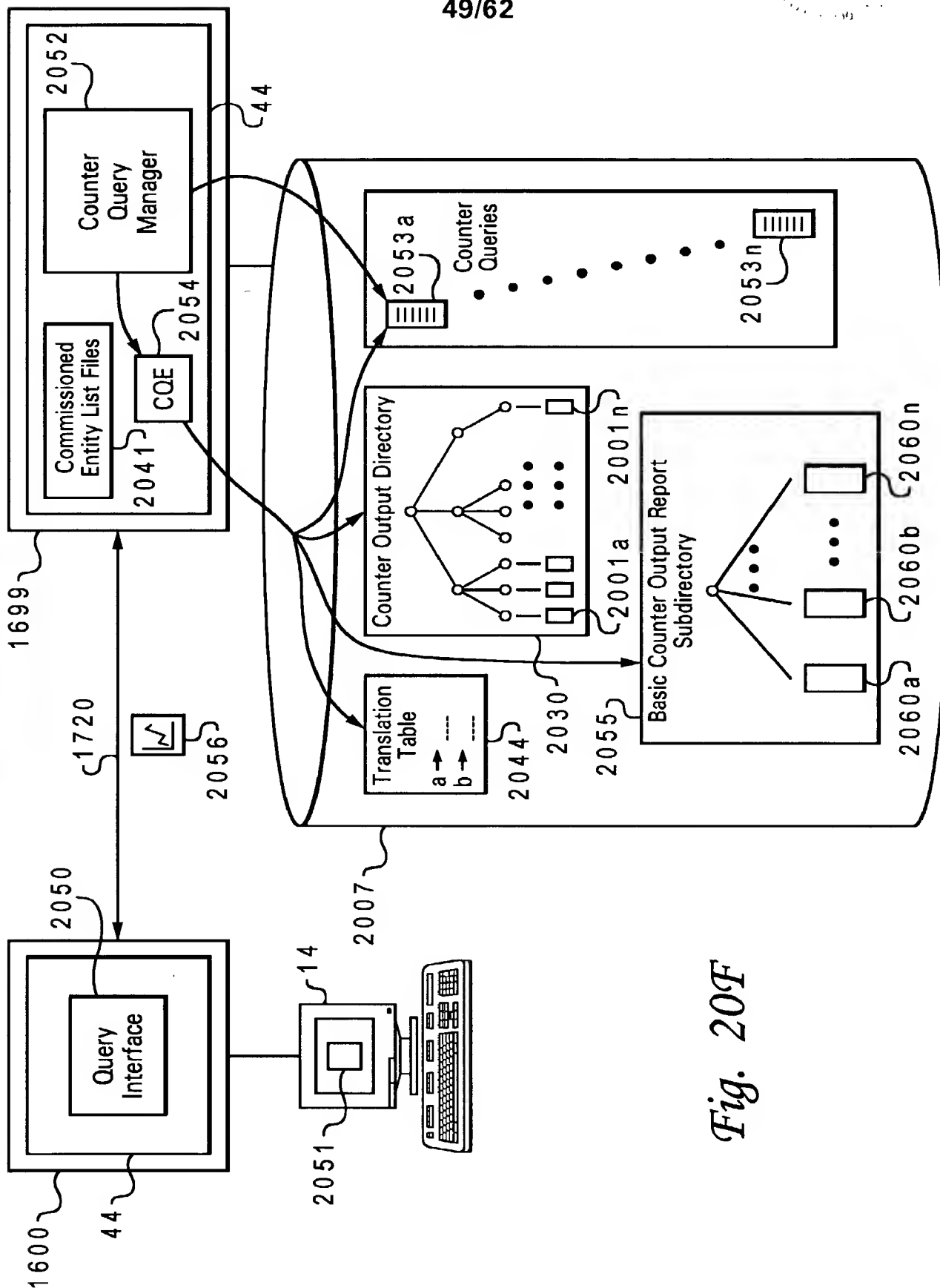
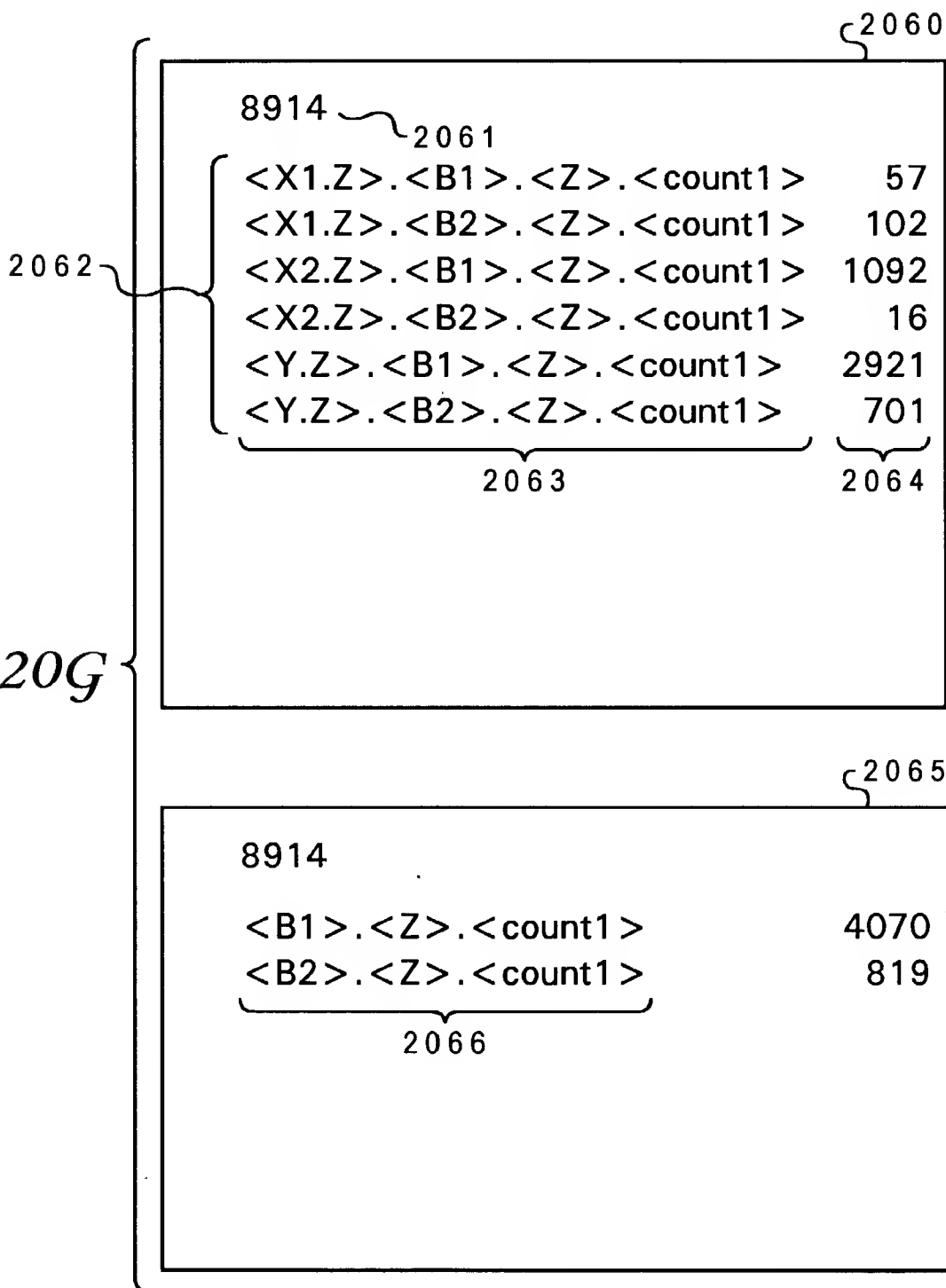


Fig. 20F

Fig. 20G



с 2053



*Fig. 20H*

52/62

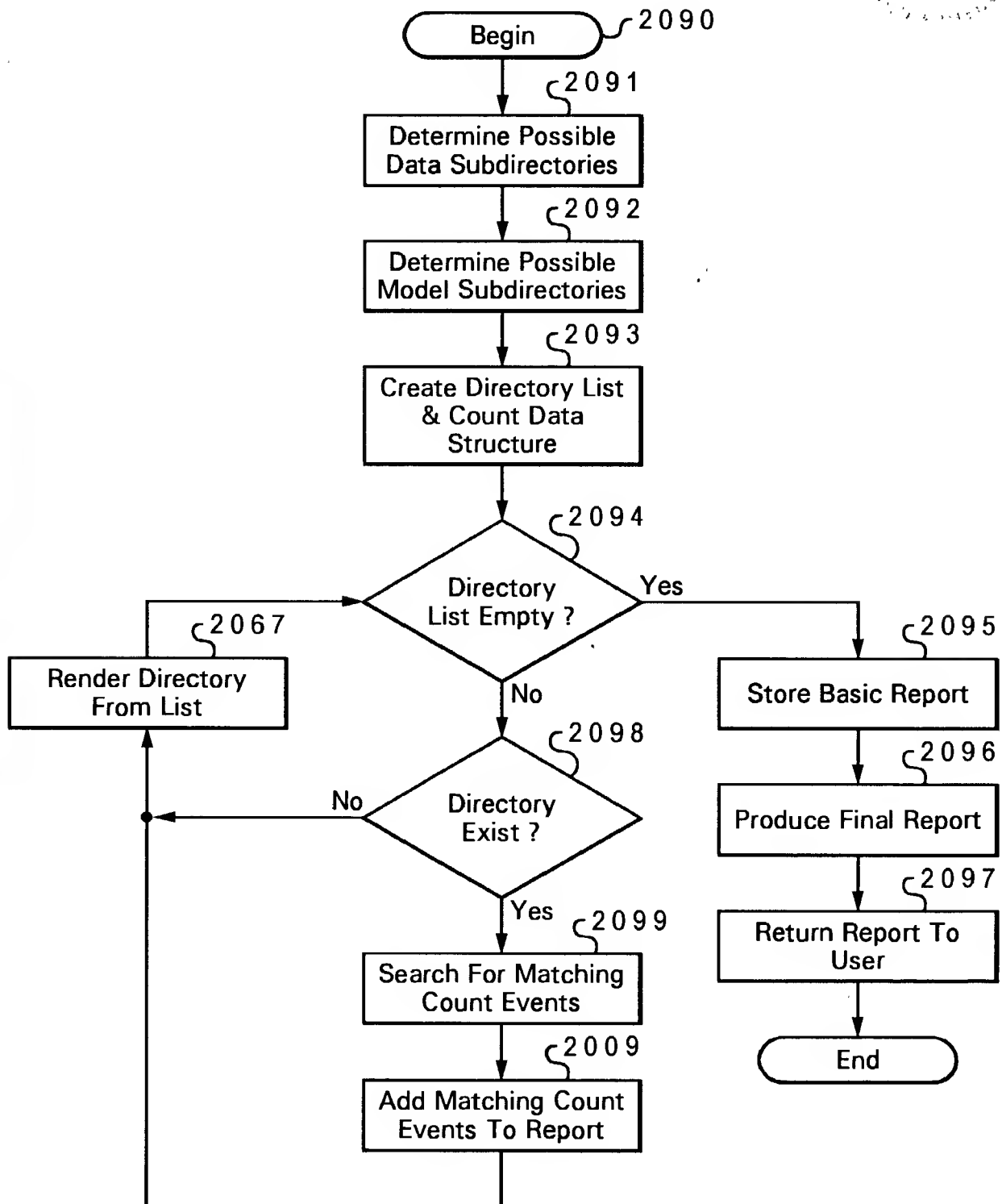


Fig. 20I

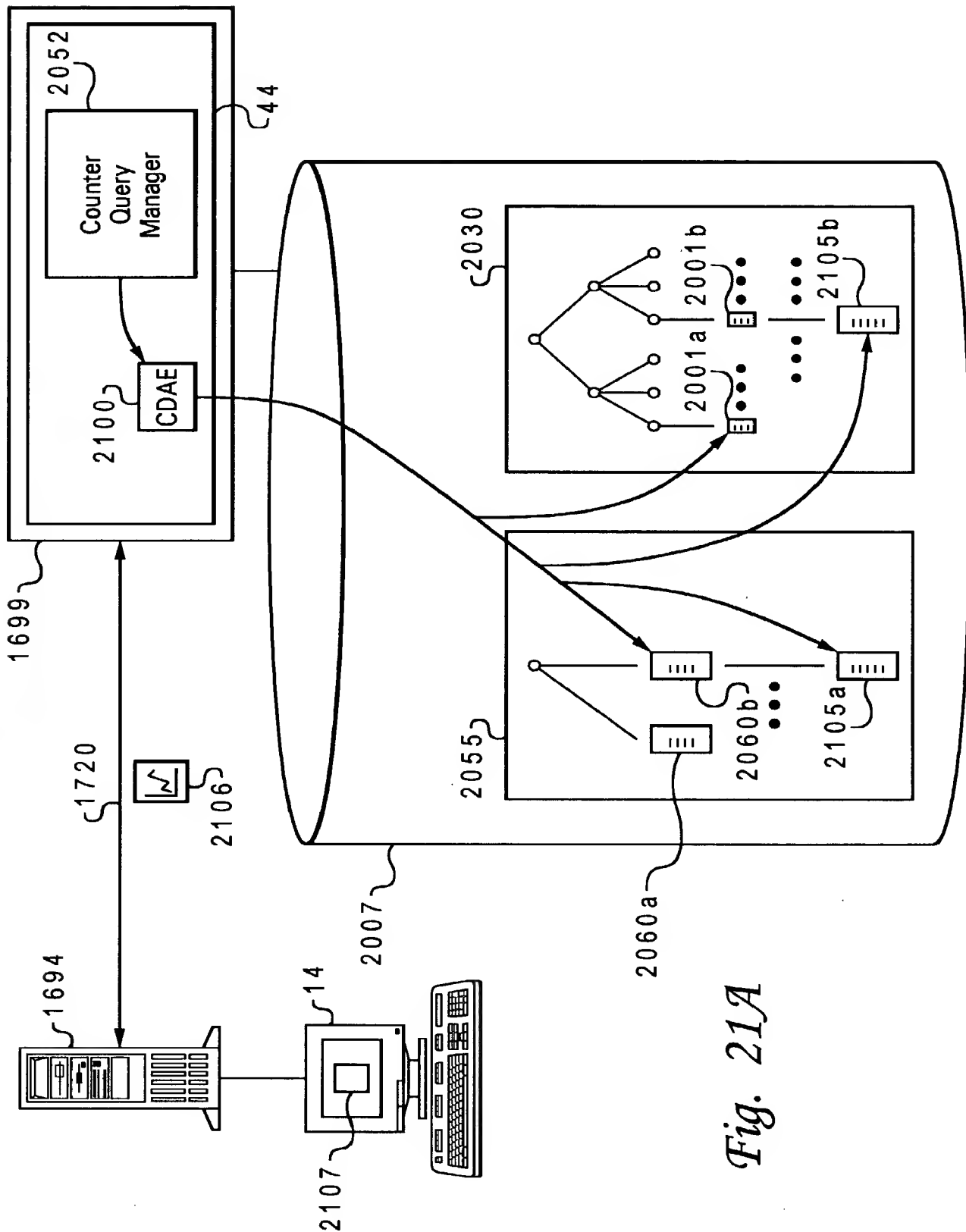
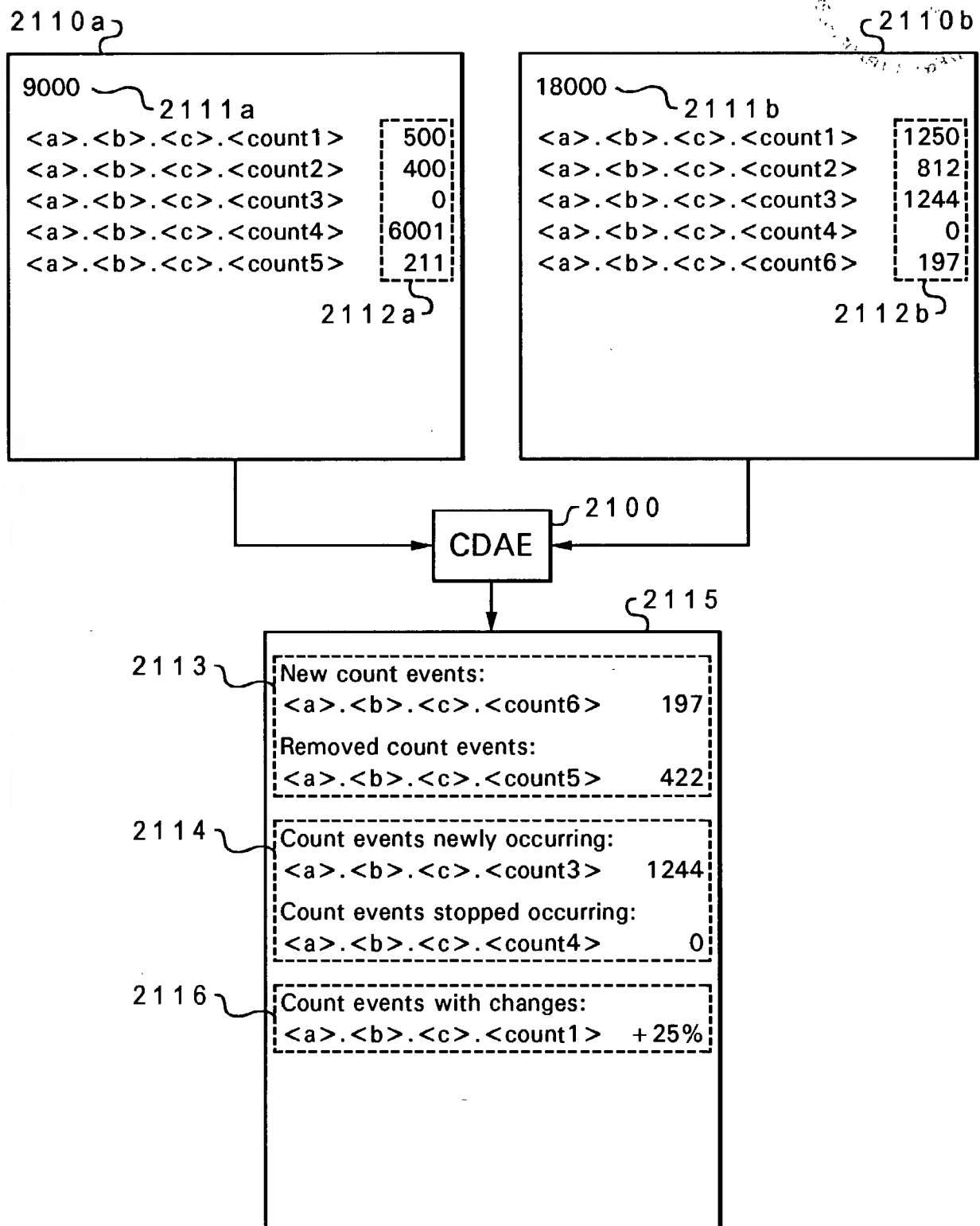
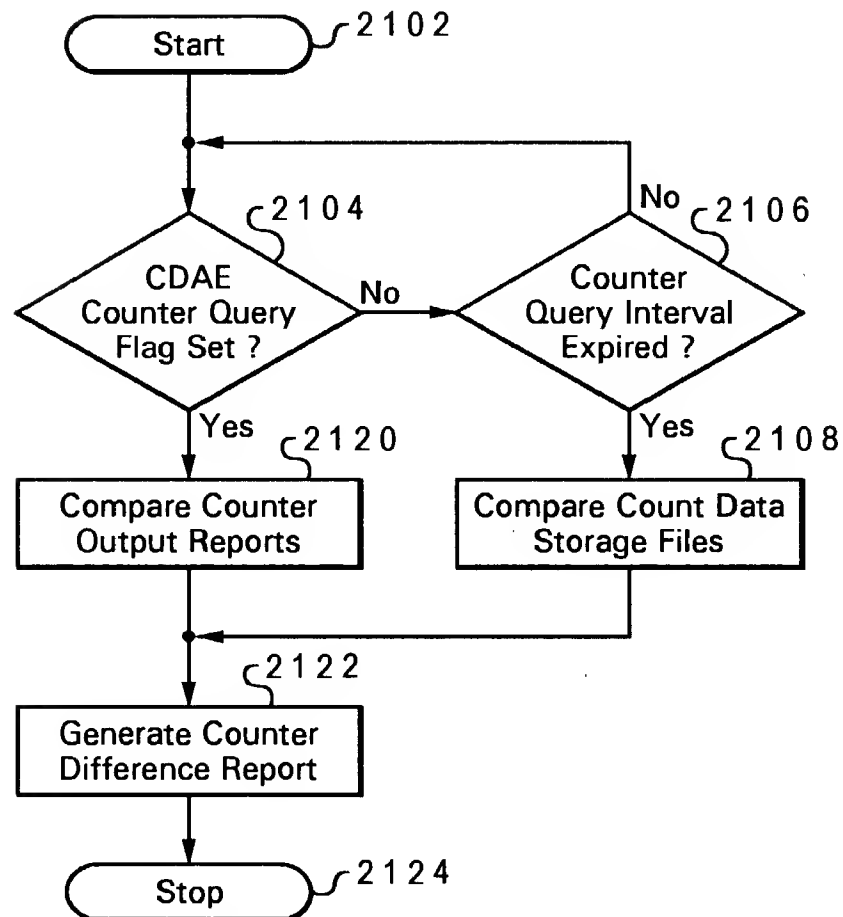


Fig. 21A

54/62



*Fig. 21B*

*Fig. 21C*

*Fig. 21D*

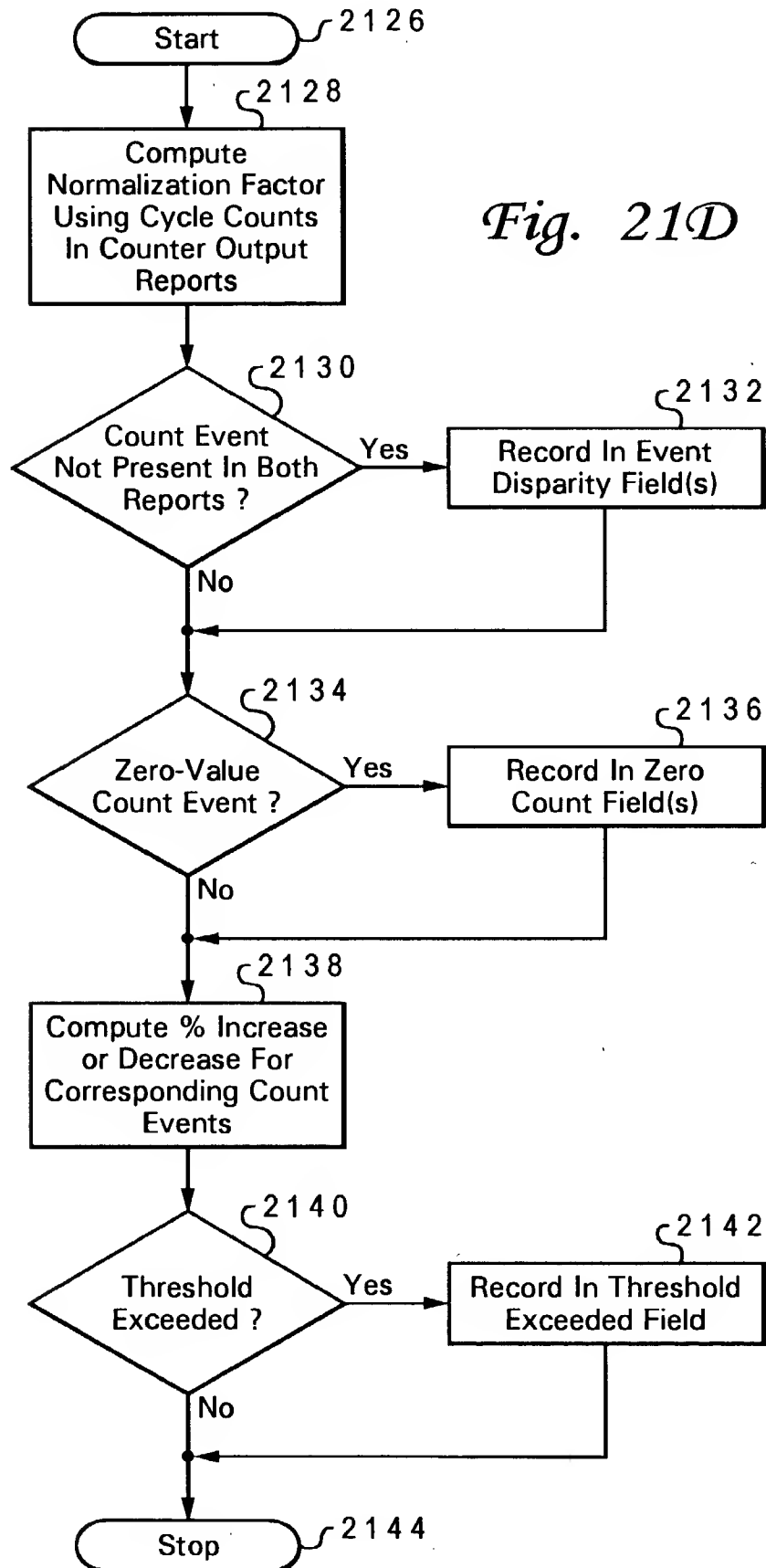






Fig. 22A

58/62

Fig. 22B

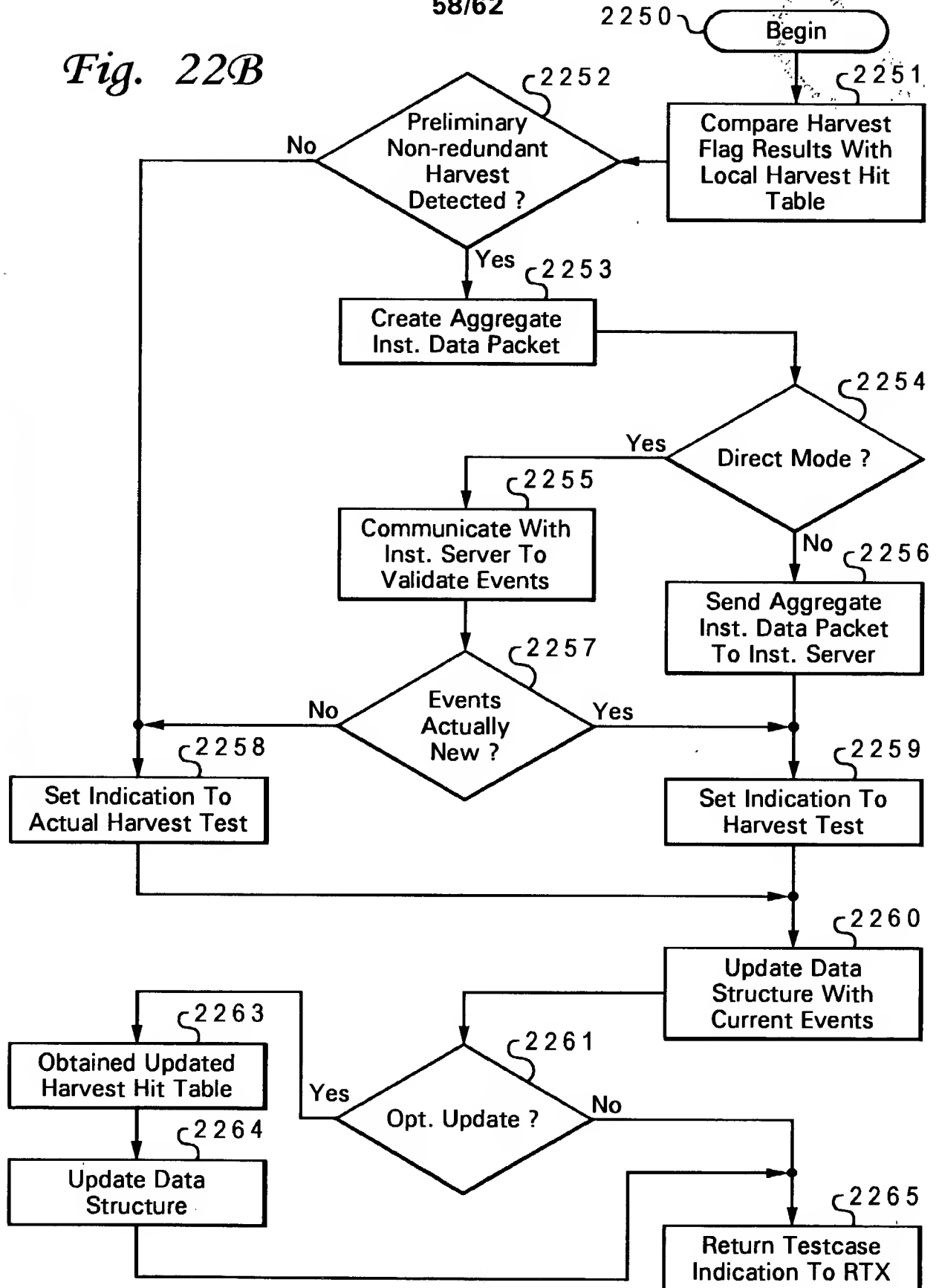
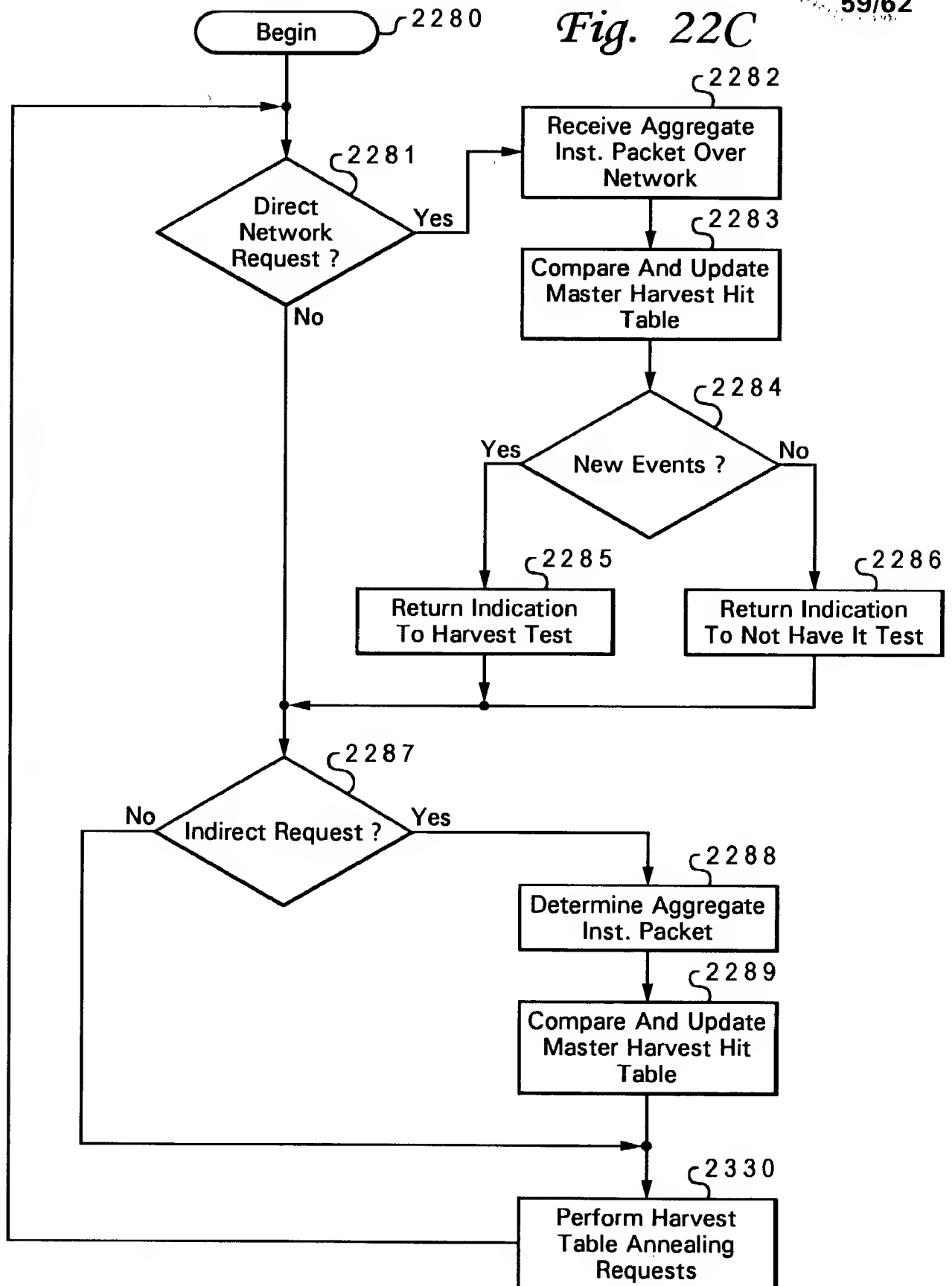


Fig. 22C



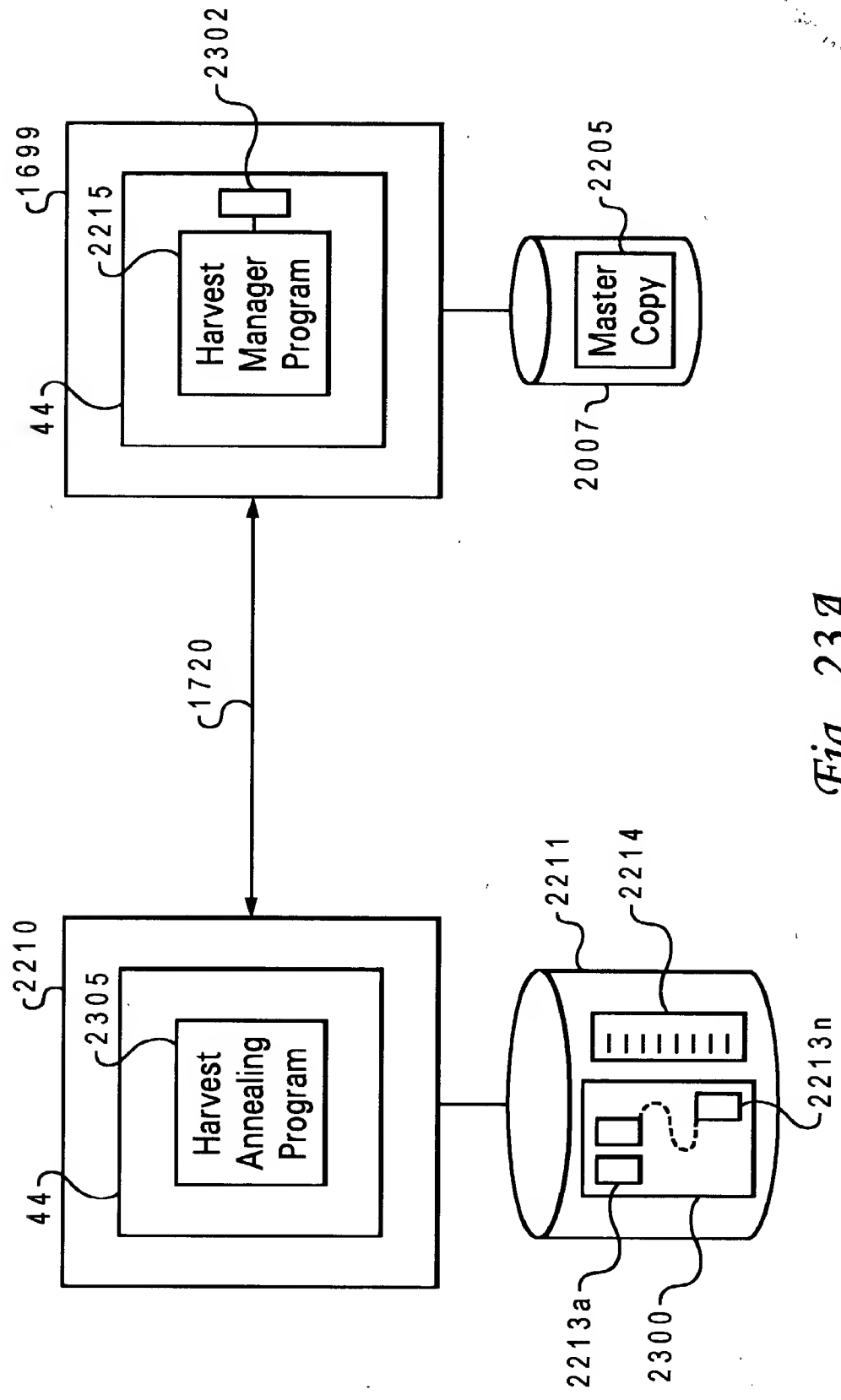
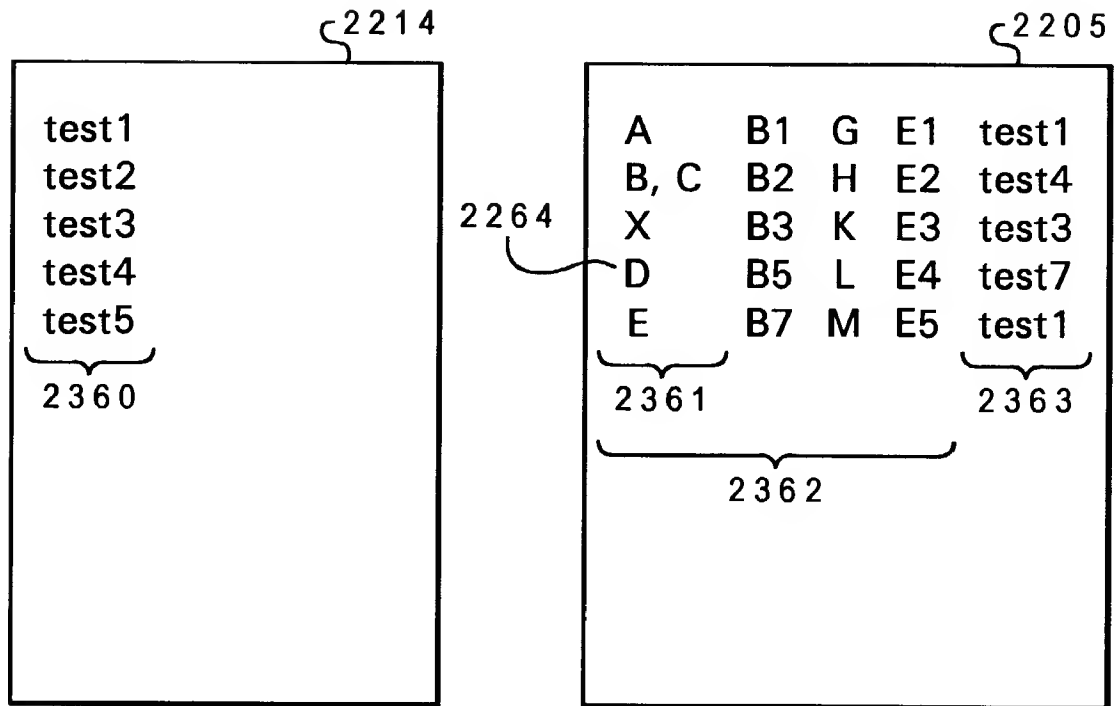
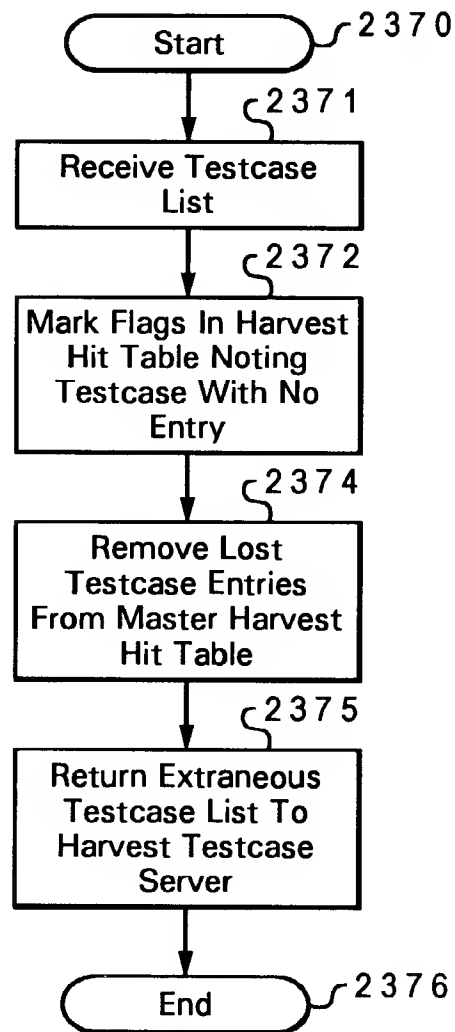


Fig. 23A



*Fig. 23B*



*Fig. 23C*